Citrix

MULTIUSER™

USER'S GUIDE

CITRIX

Citrix *MULTIUSER*

USER'S GUIDE

NOTICE

Citrix Systems, Inc., ("CITRIX"), makes no representations or warranties with respect to the contents or use of this publication, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, CITRIX reserves the right to make changes in specifications and other information contained in this publication without prior notice, and without obligation to notify any person or entity of such revisions or changes.

Patent #4,825,358
Patent #4,779,187

Citrix Systems, Inc.
210 University Drive, Suite 700
Coral Springs, FL 33071

Printed in the United States of America.

TRADEMARKS

The following names, used in this publication, are trademarks or registered trademarks:

Citrix and Citrix *MULTIUSER*	Citrix Systems, Inc.
Microsoft	Microsoft Corporation
MS	Microsoft Corporation
MS DOS	Microsoft Corporation
Codeview	Microsoft Corporation
MS OS/2	Microsoft Corporation
Microsoft Excel	Microsoft Corporation
Microsoft Word	Microsoft Corporation
IBM, PS/2, and OS/2	International Business Machines Corporation
IBM Quietwriter III	International Business Machines Corporation
PC/AT	International Business Machines Corporation
Micro Channel	International Business Machines Corporation
Presentation Manager	International Business Machines Corporation
Compaq	Compaq Computer Corporation
Hayes	Hayes Microcomputer Products Incorporated
386	Intel Corporation
National Semiconductor 16450	National Semiconductor Corporation
Esprit Opus 3n1+	Esprit Systems Incorporated
Kimtron KT-70PC	Kimtron Corporation
MC2, LINK MC2PC, and LINK MC5	Link Technologies Incorporated
IBM 3151	International Business Machines Corporation
ADDS 2025	Applied Digital Data Systems Incorporated
Wyse	Wyse Technology
Wyse 60 and Wyse 150	Wyse Technology
WY-60 and WY-150	Wyse Technology
Okidata and Okitel	Oki Electric Industry Company
Okidata Okitel 2400	Oki Electric Industry Company
Multitech and Multitech 224E	Multi-Tech Systems Incorporated
Televideo	Televideo Systems Incorporated
Televideo 965	Televideo Systems Incorporated
Boca Research IO/AT	Boca Research Incorporated
Digiboard and Digiboard PC/Xi	Digiboard Incorporated
Digiboard MC/Xi	Digiboard Incorporated
AST and AST I/O Mini II	AST Research Incorporated

TABLE OF CONTENTS

CHAPTER QUICK REFERENCE

WELCOME TO Citrix *MULTIUSER*

The Citrix *MULTIUSER User's Guide* is designed to be a task oriented, easy-to-use guide for the Citrix *MULTIUSER* system. Because Citrix *MULTIUSER* provides multiuser extensions to MS OS/2, it will be referred to in this publication as MS OS/2 *MULTIUSER.*

OVERVIEW

This guide is designed for both beginning users and those who are familiar with advanced computer topics.

In this book, you will find the following parts:

Part 1: Getting Started

Chapters in this part explain basic concepts that you will need when using MS OS/2 *MULTIUSER*:

- A visual overview of MS OS/2 *MULTIUSER.*

- Multiuser considerations.

- Learning basic MS OS/2 *MULTIUSER* skills.

Part 2: Using MS OS/2 *MULTIUSER*

Chapters in this part describe the more advanced concepts and tasks you will need to understand when you use MS OS/2 *MULTIUSER*:

- Running the command interpreter (CMD).

- Using MS OS/2 *MULTIUSER* commands and utility programs to manage and work with files, directories, and disks.

- Using MS OS/2 *MULTIUSER* multiuser features.
- Running batch files.
- Using the MS OS/2 *MULTIUSER* System Editor.

HOW TO USE THIS GUIDE

Before using the system, you should read Chapter 1, scan Chapter 2, and scan the beginning of Chapter 3 (through "Running CMD"). After doing this, you will know how to login and begin using the system.

You can then choose to read the rest of the book or use it as a reference while working at your terminal.

Before using this guide, MS OS/2 *MULTIUSER* should be installed on your system. See your System Administrator if MS OS/2 *MULTIUSER* is not installed on your system. If you have to install MS OS/2 *MULTIUSER*, refer to the *Citrix MULTIUSER System Administrator's Guide.*

Notational Conventions

To help you locate and interpret information easily, this guide uses specific typographic conventions and a standard syntax format and terminology. The following typographic conventions are used in this guide:

Text Element	Notational Convention
KEYS	Keys appear in boldface and uppercase.
variables	Variables are in lowercase italics.

Text Element	Notational Convention
USER INPUT	User input appears in uppercase and in a different typeface.
FILENAMES, *PROGRAM NAMES*, and *DEVICE DRIVERS*	Names of files, programs (including applications), and device drivers are in uppercase italics.
COMMANDS, DIRECTORY NAMES, DRIVE NAMES, and UTILITIES	These are always in uppercase.

Key Combinations

Key combinations and key sequences appear in the following format:

Notation	Meaning
KEY+KEY	A plus sign (+) between keynames means you must press the keys at the same time. For example, "Press **ALT+ESC**" means that you press the **ALT** key and hold it down while you press the **ESC** key.
KEY,KEY	A comma (,) between keynames means you must press the keys in sequence. For example, "Press **ALT**,**SPACEBAR**" means that you press the **ALT** key and release it, and then press the **SPACEBAR** and release it.

Notation	Meaning
DIRECTION keys	Arrow keys on your computer keypad indicate DIRECTION. The name refers to the direction in which the arrow on the key points: **UP**, **DOWN**, **RIGHT**, or **LEFT**.

Syntax Conventions

Syntax represents the order in which you must type a command-line command or utility name and any arguments and options that follow it. You may type commands, arguments, and options in either uppercase or lowercase letters. The following elements are used in syntax lines in this guide:

Convention	Use
KEYWORD	Elements shown in uppercase letters indicate the exact text to be entered. You may make your entry, however, either in uppercase or lowercase letters.
variables	Elements in lowercase italics are variables for which you must supply the text. For example, when *filename* appears, you should type the name of your file.
[ ]	Items in brackets are optional. To include the optional information described within the brackets, type only the information. Do not type the brackets themselves.

Convention — Use

|

A vertical bar means that you must choose from the option on either side of it. For example, ON | OFF means that you are to enter either ON or OFF.

DRIVE:

Specifies a disk drive. You need to specify a drive name along with a *filename* only if you are using a file that is **not** on the current drive. The colon (:) must be typed as shown.

PATH

Specifies a complete directory path, using the following syntax:

```
[\DIRECTORY...][\DIRECTORY...]\DIRECTORY
```

You need to specify a path along with a *filename* only if the file is not in the current directory.

...

An ellipsis indicates that an argument can be repeated as many times as necessary in a command line. Type only the information, not the ellipsis (...) itself.

PART 1

GETTING STARTED

CHAPTER 1

AN OVERVIEW OF Citrix *MULTIUSER*

INTRODUCTION

This chapter has a description of how to login and logout of the system. The operation of the Program Selector, which is what you will usually see on your display after you login, is explained. The keyboard hotkey functions are described.

THE MULTIUSER SYSTEM ENVIRONMENT

Introduction

Most MS OS/2 *MULTIUSER* systems are set up to require you to login before you can interact with the system. After you login, the system ensures that you get the operating environment that was set up for you. Part of that operating environment is usually an initial display that contains the Program Selector. The Program Selector can be set up to allow you to select a program to run from a menu of many programs.

Some systems may be set up so you don't have to login. These systems may be set up to run one or several specific programs. The systems can also be set up so your first display is a specific program or the command prompt instead of the Program Selector. The command prompt is controlled by the command interpreter (CMD) which is described in Chapter 3.

The difference between using MS OS/2 *MULTIUSER* and using a standalone personal computer is that when you are using MS OS/2 *MULTIUSER*, other people may be using the same computer and disk drive at the same time that you are. Chapter 2 provides more information on many of the multiuser aspects of the system.

Terminals

You interact with the MS OS/2 *MULTIUSER* system at your terminal. A terminal consists of a keyboard, a display, and a communications connection to the host computer. If you have experience using a personal computer, you will notice that using a terminal with MS OS/2 *MULTIUSER* feels virtually the same as using a personal computer. MS OS/2 *MULTIUSER* supports a wide range of terminals. See your System Administrator if you believe you need any specific information about the terminal you are using.

System Console

With MS OS/2 *MULTIUSER*, the system console usually works the same way as any other terminal. The system console is physically part of the host computer. The host computer must be powered on for the MS OS/2 *MULTIUSER* system to function. If the host computer is powered off, all the terminals connected to the system will stop functioning and anything that was not saved will be lost.

Normally, the System Administrator configures the keyboard hotkeys (certain special keystroke combinations) of the system console to function the same way that they will at all the terminals. Under certain problem determination situations, the System Administrator may configure certain keyboard hotkeys at the system console to operate differently than they do at all the other terminals. See the description of the "Restart Terminal Hotkey" in this chapter if you need more information.

HOW TO OPERATE YOUR TERMINAL

Introduction

Your System Administrator will set up your terminal before you operate it. You will be given a *username* and, if necessary, a *groupname* and *password* to use when you login. A *loginname* consists of at least a *username* and, depending on how your system is configured, there may also be a *groupname.* This appears in the following manner throughout this guide:

loginname = *username*[*.groupname*]

Turning Your Terminal On

Because MS OS/2 *MULTIUSER* supports several terminals and the terminal ON/OFF switches are located in different places, refer to your terminal owner's manual for switch locations. The locations of the controls for some terminals can be found in Appendix A.

Turn your terminal on. If your terminal is not set for autologin, a Login: prompt appears in the top left corner of the screen. If you do not see this, press the **ENTER** key on your terminal. If nothing happens or you see garbled data, contact your System Administrator.

HOW TO LOGIN

Introduction

MS OS/2 *MULTIUSER* has a login procedure to keep track of who is authorized to use the system. This login procedure uses a *loginname* to identify you, and optionally a *password* to check whether you are granted access to the system.

Your System Administrator grants you access to the system by giving you a *loginname* and optionally a *password.* You use the *loginname* and *password* when a Login: prompt appears on your terminal. A terminal may display the Login: prompt when it is powered on.

The System Administrator may also configure a terminal to go directly to an application when it is powered on instead of displaying a Login: prompt. In this case, your *loginname* automatically grants you access to the system when the terminal is powered on.

Username, Groupname, Loginname

When you start using MS OS/2 *MULTIUSER*, your System Administrator assigns a *loginname* to you. A *loginname* consists of a *username* and a *groupname.* Your *username* tells the system to assign resources to you and permits you to work in the system in accordance with your security level. Your *groupname* gives you a group classification and further defines what resources you have available and what your security classification is.

You use your *loginname* when you see a Login: prompt at a terminal. Enter your *username* and *groupname* and separate the two with a period. For example, if your *username* is USER1 and your *groupname* is ACCOUNTING, you would enter USER1.ACCOUNTING as your *loginname.*

Depending on your activities, the System Administrator may assign several *loginnames* to you. For example, if you are the manager of an accounting department, you might be assigned two *loginnames,* USER1.ACCOUNTING and USER1.MANAGER. The *loginname* USER1.ACCOUNTING gives you access to the accounting applications and data. The *loginname* USER1.MANAGER gives you access to personnel related databases that require restricted access.

If you have only one *loginname*, you only need to enter your *username* at the Login: prompt and the system uses your *username* and your default *groupname* as your *loginname*. Your default *groupname* is set up when the System Administrator configures your user profile.

If you have several *loginnames*, one *username* and several *groupnames*, the System Administrator has configured your user profile so that one of your *groupnames* is your default *groupname*. When you enter your *username* at the Login: prompt, the system uses your *username* and your default *groupname* as your *loginname*. For your other *groupnames*, you must enter the full *loginname*, both the *username* and *groupname*.

Password

The System Administrator may assign a *password* to use with your *loginname*. The *password* provides an extra measure of security, since *loginnames* of authorized users can become known to others seeking access to the system. You should follow any guidelines set up by your System Administrator to protect your *password*.

If you forget your *password*, or have difficulty in logging in, contact the System Administrator. The password is case sensitive; hence, password ABCDEF is not the same as abcdef or ABCdef.

Refer to the section "Changing Your Password" in Chapter 5 for information about using the PASSWORD program to change your *password* as well as to obtain information about your *password*.

Login Example

To login, do the following:

1. Turn your terminal on. The Login: prompt appears on your screen similar to:

   ```
   Login:
   ```

2. Enter your *username* and *groupname*, and then press **ENTER**. In the example below, USER1 is the *username* and ACCOUNTING is the *groupname.*

   ```
   Login: USER1.ACCOUNTING
   ```

3. The *password* prompt appears. Type your *password,* and then press **ENTER**. Your *password* is not displayed as you type it. Instead, an underscore (_) appears for each character you type. If you do not have a *password,* just press **ENTER**.

   ```
   Login: USER1.ACCOUNTING
   Password: _ _ _ _ _ _
   ```

   Your initial program appears. This may be the Program Selector, CMD command prompt, or an application.

4. If you do not enter the correct *username, groupname,* or *password,* the message "Login failed" appears on your screen. Re-enter your *username, groupname,* and *password.* If the message reappears, contact your System Administrator.

Autologin Feature

Your System Administrator may configure a terminal so you do not have to login. In this case, the login procedure is performed automatically at this terminal. This terminal uses the autologin feature and is known as an autologin terminal.

Depending on how the autologin terminal is configured, the Program Selector, the command prompt, or an application may appear when you power it on. You can immediately begin using the terminal.

Logging In After Being Disconnected

The System Administrator may configure your *loginname* with the auto-disconnect feature so that when your terminal loses power unexpectedly or times out, the system automatically disconnects your login from the terminal instead of terminating your login. A disconnected login continues to run in the background. To reconnect to your login, you must perform the login procedures. The system matches the disconnected login to your *loginname* and reconnects the login to the terminal where you are logging in, even if that terminal is different from the terminal where you originally logged in.

The System Administrator may configure the system so that your login must reconnect to the same terminal where you originally logged in. If this is the case, you must perform the login procedures at the same terminal that you used before.

You may explicitly disconnect your login with the DISCONN utility. This is useful if you need to leave your terminal but wish to keep the applications running. Refer to Chapter 5, "Using Disconnected Logins."

HOW TO LOGOUT

Introduction

If you are using an application at an autologin terminal, you should follow the application's instructions for ending your work with the application. Your System Administrator may instruct you on whether you should end the application and power off the terminal.

If you are using an application at a terminal that has the Login: prompt when you power on, there are two methods to terminate your login. The recommended method is to switch to each application running in the login and end the application according to the application's instructions. This allows the application to save data that it is using. For the command interpreter, type EXIT at the command prompt. When the last program in the login has terminated, the login is automatically logged out. If you are using the Program Selector, use the Program Selector logout function after terminating all your applications. See "Logging Out from the Program Selector" later in this chapter.

Another method to terminate your login is to use the LOGOUT program. The LOGOUT program forces all applications in your login to terminate. However, since the application is being forced to end, it may not be able to save data that it is using.

Logout Examples

1. If you are using the Program Selector, refer to Chapter 1, "Program Selector, Quitting Applications." This section explains how to switch to applications using the Program Selector. Follow each application's instructions for ending each application.

For example, CMD (the command interpreter) uses the EXIT command to terminate. Therefore, if you type EXIT at the command prompt, the command interpreter terminates and the session ends.

After you have ended all the applications, refer to Chapter 1, "Program Selector, Logging Out from the Program Selector."

2. If you are not using the Program Selector, follow the instructions of the application you are using to terminate the application. If you have another application running in the background, it becomes foreground when the first application terminates. Terminate this application according to its instructions. To terminate the command interpreter, type EXIT at the command prompt. When you have terminated all the applications in your login, your login is automatically logged out.

IMPORTANT KEYBOARD CONSIDERATIONS

Introduction

This section describes some keyboard hotkeys MS OS/2 *MULTIUSER* supports. Hotkeys are certain key combinations that, when pressed together, cause the system to perform an action.

Terminal HotKeys

Each MS OS/2 *MULTIUSER* terminal can be running more than one application in different screen sessions. The System Administrator, using the terminal configuration command CONFIG TERMINAL, determines the hotkey support that is available for each of the MS OS/2 *MULTIUSER* terminal configurations. MS OS/2 *MULTIUSER* supports many kinds of terminals. The default hotkey key combinations may be customized by the System Administrator to accommodate a particular terminal that does not have one or more of the keys.

See the following table for an overview of the terminal hotkey support.

Hotkey	Function	Description
ALT+ESC	Session Switch	Switches to the next session (program) running under the current login.
CTRL+ESC	Session Direct	Switches the Program Selector to the foreground.
ALT+TAB	Session Create	Creates a new session running the program specified by the COMSPEC environment variable, which defaults to CMD.
CTRL+TAB	Login Switch	Switches to the next login at the current terminal.
CTRL+C or **CTRL+Break**	Stop Program	Stops the program you have running.
PRINT SCREEN	Print Screen	Sends the screen image of the current foreground session to the printer.
CTRL+PRINT SCREEN	Print Screen Toggle	Echoes future output to the printer until this hotkey is pressed again.
CTRL+ALT+DEL	Restart Terminal	Terminates all applications running on the terminal, then restarts the terminal.

The keyboard technique and details for each of the terminal hotkey functions are as follows:

- Session Switch Hotkey

  To switch between your sessions (programs) running under the current login, press **ALT+ESC.** If you press **ALT+ESC** repeatedly, you switch between all of the programs running under the current login.

- Session Direct Hotkey

  To switch to the Program Selector, press **CTRL+ESC**. The Program Selector displays the applications you are currently running. However, an application you are using can change the behavior of this hotkey.

- Session Create Hotkey

  To create a new session running the command interpreter, showing the command prompt, press **ALT+TAB**. It is possible to configure the system so that a different program is started in the new session.

- Login Switch Hotkey

  More than one login can be concurrently active at the same terminal. To switch between active logins on a terminal, press **CTRL+TAB** until you reach your login. See "Using Disconnected Logins" in Chapter 5 for more information.

- Stop Program Hotkey

  To stop the program you have running, press **CTRL+C** or **CTRL+BREAK**.

- Print Screen Hotkey

  To print the screen image of the current foreground session, press **PRINT SCREEN.**

- Print Screen Toggle Hotkey

  To echo your future command line output to the printer, press **CTRL+PRINT SCREEN**. Press **CTRL+PRINT SCREEN** again to stop echoing command line output to the printer. All your command line output, including characters you type at the command prompt and error messages generated during that interval, are sent to the printer.

- Restart Terminal Hotkey

  At a terminal other than the console, to terminate all applications running on the terminal and restart the terminal, press **CTRL+ALT+DEL**.

  At the system console, if REBOOT=ON is specified in the *CONFIG.SYS* file, **CTRL+ALT+DEL** terminates all applications running on the system and restarts the system. If REBOOT=OFF, the Restart Terminal Hotkey terminates only the console applications and restarts the console terminal.

  The SHUTDOWN command may also be used to terminate all system applications. For additional information on REBOOT and SHUTDOWN, see the *Citrix MULTIUSER Command Reference.*

FIRST USER PROGRAM

The MS OS/2 *MULTIUSER* System Administrator determines which screen you see first after you login. You may see the Program Selector, the CMD command line prompt, or go directly to an application.

PROGRAM SELECTOR

Introduction

The Program Selector provides an easy interface for starting and switching between applications. It consists of two menus: the Start Programs menu and the Task Selector menu.

The Start Programs menu enables you to:

- Start an application on the list.
- Get access to a command prompt by starting a command interpreter session.
- Add, delete, or update programs in the Start Programs menu list.
- Logout.

The Task Selector menu enables you to:

- Switch to a program already running.
- End a program that is running.

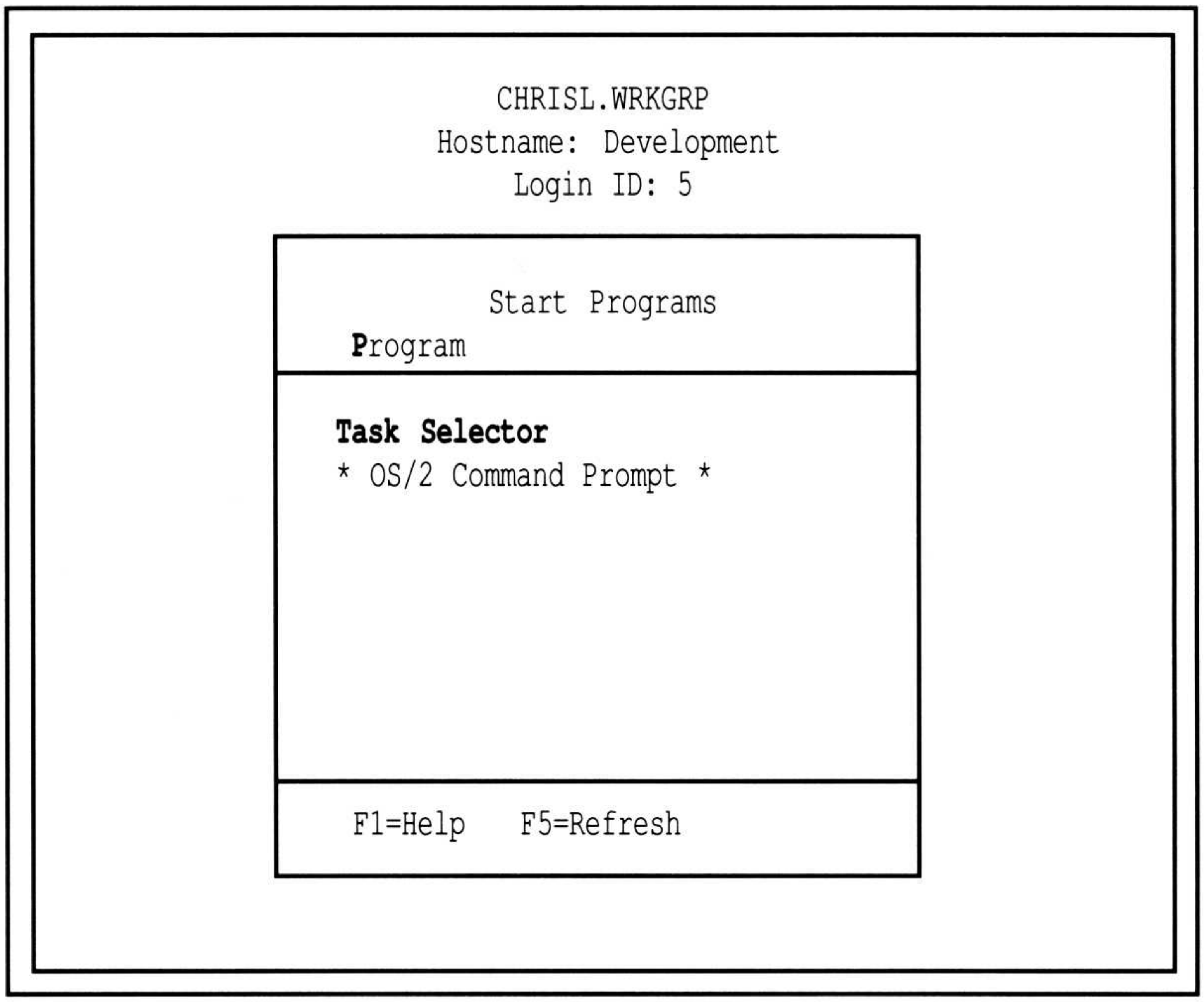
NOTE: The **CTRL+ESC** hotkey allows you to switch from an application to the Program Selector. Once you have the Program Selector menus on the screen, you can press **ESC** to switch between the Task Selector and the Start Programs menus.

First Time you Login

The first time you login, you may need to create the Program Selector files that store the selections in the Start Programs menu. If these files do not exist, you will get a warning message informing you that "*PSEL01.AIF*" was not found. Press **ENTER** and a default file will be created.

Start Programs Menu

The Start Programs menu lets you choose a program to run from a list of programs that are installed on your system. It also has a pull down menu for working with your Start Programs list. You can also select the Task Selector menu where you can switch to running a program. The Task Selector menu is described later in this chapter.



```
                    CHRISL.WRKGRP
                 Hostname: Development
                     Login ID: 5

            Start Programs
  Program

 Task Selector
 * OS/2 Command Prompt *

  F1=Help    F5=Refresh
```

Selecting a Program

To select a program, use your **DIRECTION** keys to move the selection bar over the program name you want to start. Press **ENTER** and the program will begin to run.

Selecting the Program Pull Down Menu

Move the selection bar over the program name you want to work with. Press **F10** to move the cursor to the action bar. Press **ENTER**. The following selections appear:

Start
: Starts the program that was highlighted with the selection bar.

Add
Adds a program to the Start Programs menu.

Update
Modifies the program that was highlighted with the selection bar.

Delete
Deletes the program that was highlighted with the selection bar.

Logout
Ends your user login.

Use the **DIRECTION** keys to move the selection bar over the function you want and press **ENTER**.

Press **ESC** to return to the previous menu.

Selecting Options

Selection Bar

When an MS OS/2 *MULTIUSER* menu, such as the Program Selector's Start Programs menu appears, use the **DIRECTION** keys to move the selection bar up and down in the menu. The selection bar highlights the choices available. Once you have highlighted your choice, press **ENTER** to execute your selection.

Start Programs Function Keys

When a menu appears and one of the letters (usually the first letter) of each of the menu choices are highlighted, you can select the choice you want by pressing its highlighted letter on the keyboard.

When you are at the Start Programs or Task Selector menu, selected keys will have the following functions:

ALT+F6

Toggles to/from the help window and the menu selection if it is not overlaid by the help window.

DIRECTION KEYS

Move the selection bar among menu choices or data entry fields.

ENTER

Executes the highlighted menu choice or saves information typed in data entry fields.

ESC

Switches between the Start Programs menu and Task Selector menu, or cancels the current operation and returns to the previous menu.

F1

Displays HELP about the field where the selection bar is currently located.

F5

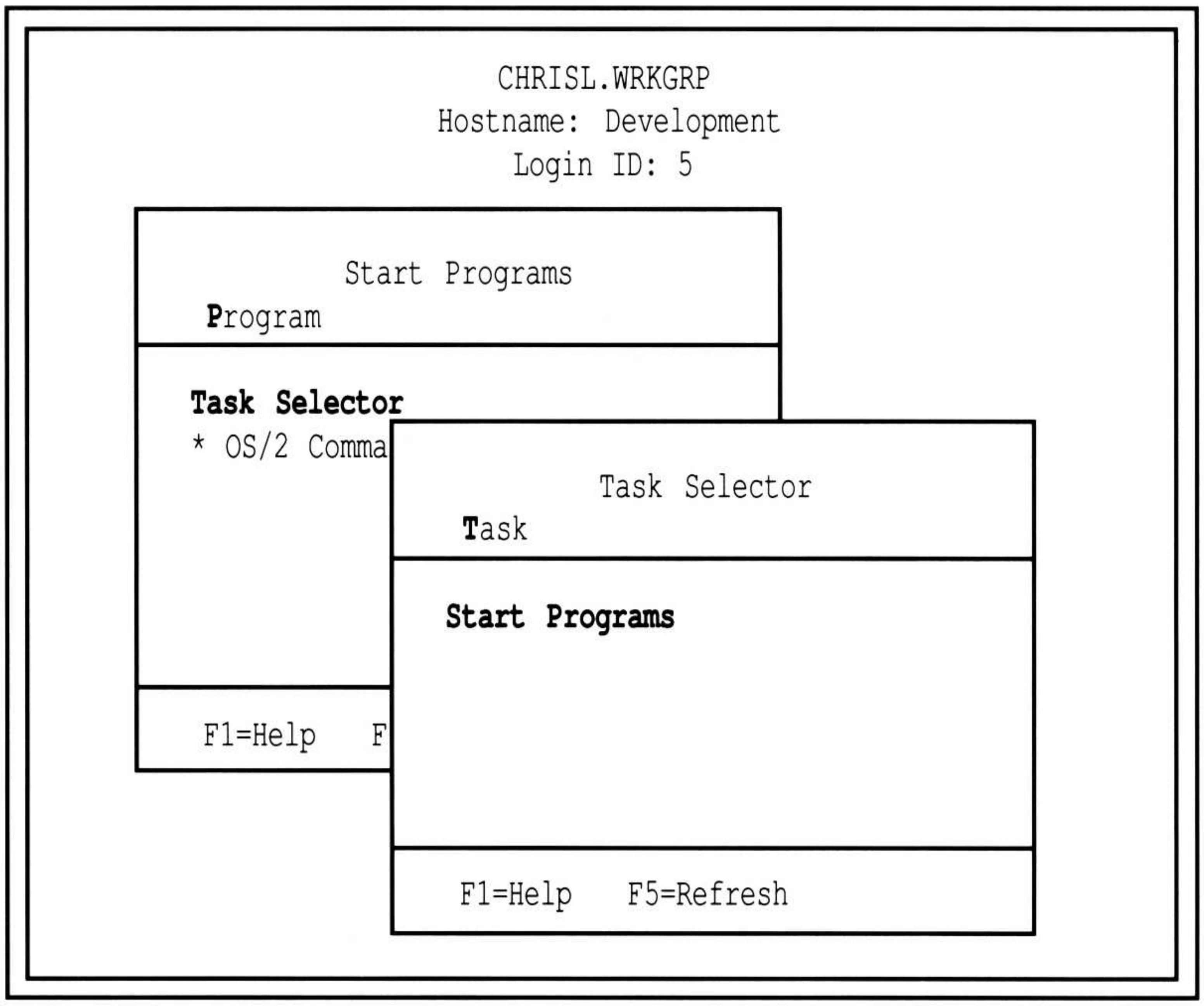
Refreshes the menu.

F10

Toggles the cursor between the action bar at the top of a window and the menu selections in the body of the window. **NOTE**: The **ALT** key performs the same action.

Task Selector Menu

The Task Selector menu displays a list of programs that are currently running. From the Task Selector menu you can switch to one of the currently running programs. It also has a pull down menu for switching to and closing running programs.



```
CHRISL.WRKGRP
Hostname: Development
Login ID: 5

Start Programs
Program

Task Selector
* OS/2 Comma

F1=Help   F

Task Selector
Task

Start Programs

F1=Help   F5=Refresh
```

To go to the Task Selector menu, move the selection bar over the "Task Selector" line in the Start Programs menu and press **ENTER**. You can also press **ESC** to go to the Task Selector menu.

Selecting a Program

To select a currently running program, move the selection bar over the program name you want to switch to and press

ENTER. This brings the selected program into the foreground.

Selecting the Task Pull Down Menu

Move the selection bar to the program you want to switch to or close. Press **F10** to move the cursor to the action bar. Press **ENTER**. The following selections will appear:

Switch to
Switches you to the program that was highlighted with the selection bar.

Close
Closes the program that was highlighted with the selection bar. **NOTE**: When a program is closed, all unsaved data will be lost.

Press **ESC** to return to the previous menu.

Task Selector Function Keys

When you are at the Task Selector menu, selected keys will have the same functions as those described in the section "Start Program Function Keys."

Adding an Application to the Start Programs List

Use the following procedure to add an application to the Start Programs list:

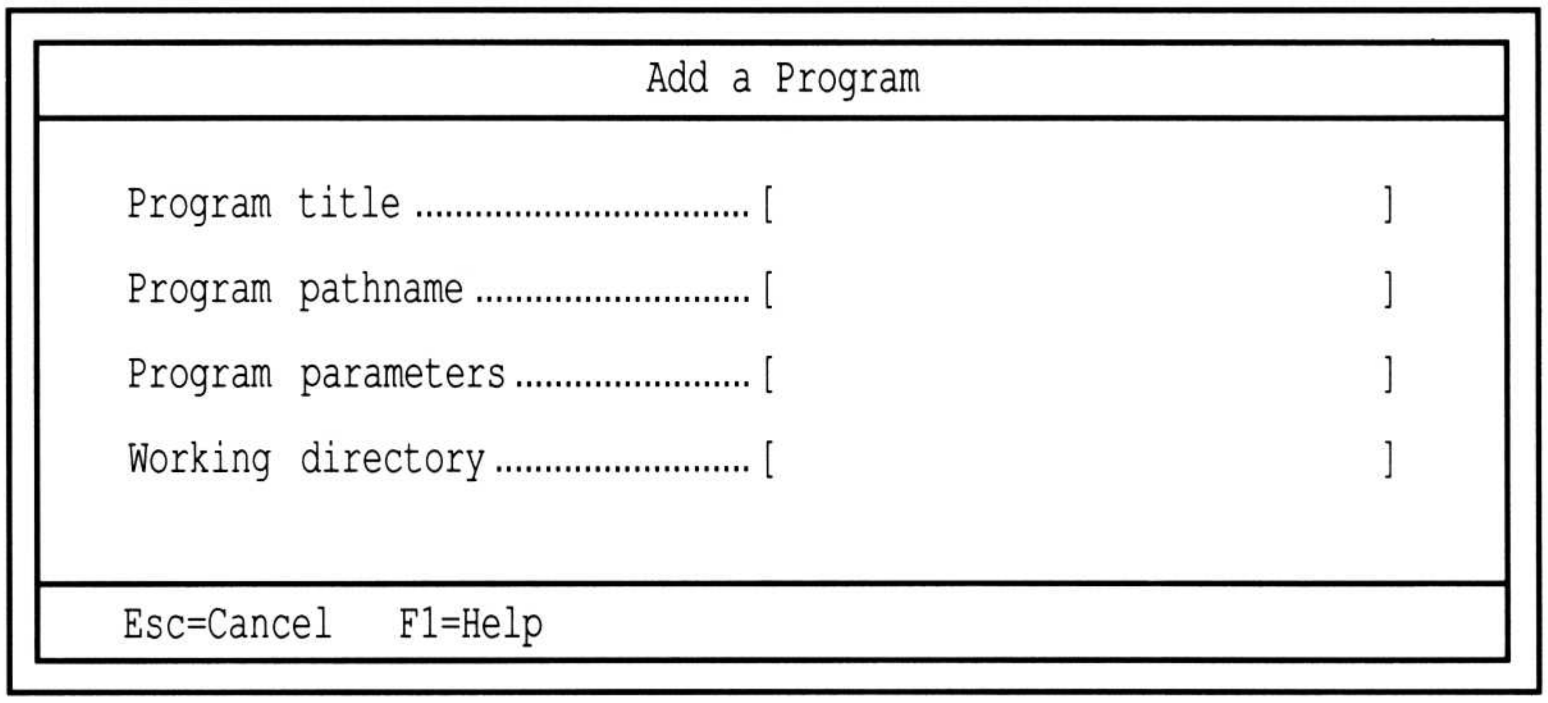
1. From the Start Programs menu, press **F10** to go to the action bar.

2. Select Program and press **ENTER**.

3. Select Add and press **ENTER**.

4. Fill in the fields illustrated in the figure below. If you need help on a field, move the cursor to the field and press **F1**.

NOTE: Leaving the Program pathname field blank will start the command interpreter.



```
                         Add a Program

  Program title ..................[                         ]
  Program pathname ...............[                         ]
  Program parameters .............[                         ]
  Working directory ..............[                         ]

  Esc=Cancel   F1=Help
```

5. Press **ENTER**.

The application is displayed in the Start Programs menu for you to select and start.

Starting an Application

Use the following steps to start an application already installed on the system for your use:

1. Login to the system
   a. Enter your *username* and *groupname*
   b. Press **ENTER**
   c. Enter your *password*
   d. Press **ENTER**

   The Program Selector appears if it is configured as your first user program.

2. Go to the Start Programs menu of the Program Selector.

3. Select the application
   a. Use your **DIRECTION** keys to move the selection bar over the application you want to start.
   b. Press **ENTER**

   Your application will begin.

Switching Among Applications

Because you can run several applications at the same time under MS OS/2 *MULTIUSER*, you may need to switch among applications. Each application runs in its own session. Use the following procedure to switch among applications:

1. Call up the Task Selector menu that lists all of the currently running applications.
   a. Press **CTRL+ESC** to bring the Program Selector to the foreground.

   or

   b. Use your **DIRECTION** keys to highlight the Task Selector line in the Start Programs menu and press **ENTER**.

2. Use your **DIRECTION** keys to highlight the application you want to switch to and press **ENTER**.

NOTE: You can also switch among applications (sessions) by pressing **ALT+ESC**.

Quitting Applications

You can quit applications in more than one way. The major difference between the ways of quitting an application is whether or not data will be preserved.

Quitting Applications and Preserving the Data

1. Call up the Task Selector menu that lists all of the currently running applications.
   a. Press **CTRL+ESC** to bring the Program Selector to the foreground.

   or

   b. Use your **DIRECTION** keys to highlight the Task Selector line in the Start Programs menu and press **ENTER**.

2. Select the application you want to quit. Use the **DIRECTION** keys to move the selection bar over the application and press **ENTER**.

3. Use the command that the application uses for a safe shutdown. Refer to the application's manual.

4. Follow the same procedure for other applications you want to quit.

Quitting Applications Without Preserving Data

1. Call up the Task Selector menu.
   a. Press **CTRL+ESC** to bring the Program Selector to the foreground.

   or

   b. Use your **DIRECTION** keys to highlight the Task Selector line in the Start Programs menu and press **ENTER**.

2. Select the application you want to quit
   a. Use the **DIRECTION** keys to move the selection bar over the application.

   b. Press **F10** to move the cursor to the action bar.

   c. Press **ENTER** and the Task menu appears.

   d. Use the **DIRECTION** keys to move the selection bar over Close and press **ENTER** or press **C**. A warning message appears. Press **ENTER**. The application is closed. Unsaved data will be lost.

   e. Repeat the steps for the other applications you want to quit.

Logging out From the Program Selector

At the Start Programs menu, press **F10** to move to the action bar and press **ENTER**. A menu will appear. Move the selection bar over Logout and press **ENTER** or press **L**.

NOTE: Properly quit your applications before logout or any unsaved data will be lost.

Startup File (*STARTUP.CMD*)

The *STARTUP.CMD* is the first program executed by the Program Selector. *STARTUP.CMD* is a batch file used to start other sessions and it is executed whenever the Program Selector is started. For more information, refer to "Using Startup Files" in Chapter 6.

HOW TO GET HELP

MS OS/2 *MULTIUSER* utilities and the Program Selector contain help to answer questions you might have. If the help information does not answer your questions, refer to the *Citrix MULTIUSER System Administrator's Guide* or the System Administrator.

If you are running the Program Selector or an MS OS/2 *MULTIUSER* utility that has help messages, press **F1** to display help. Most of the MS OS/2 *MULTIUSER* utilities with multiuser features have help for their command line parameters. You can display the command line parameters for these utilities by typing the utility name at the MS OS/2 *MULTIUSER* command prompt followed by **/?**. Press **ENTER**. If the utility supports command line help, you see a list of the command line parameters supported by the utility.

You can also get help for MS OS/2 *MULTIUSER* messages. Refer to the "Getting Help" section of "Running CMD" for information on message help.

Refer to your application's manuals for help with an application.

CHAPTER 2

MULTIUSER CONSIDERATIONS

INTRODUCTION

As we discussed at the beginning of Chapter 1, many other people may be sharing the same computer and devices with you at the same time. This is why MS OS/2 *MULTIUSER* makes a number of multiuser features available to your System Administrator for configuring, operating, and maintaining the MS OS/2 *MULTIUSER* system. These features allow the MS OS/2 *MULTIUSER* system to be set up in such a way that potential interference between multiple users is minimized or completely eliminated. This chapter reviews some of the security and resource management features that provide this capability.

For example, if you are familiar with the operation of a standalone personal computer, you are used to having complete access to the entire fixed disk and your initial default directory is normally the root directory of your fixed disk. Because this would not be practical in a multiuser environment, the system is normally set up so each user gets a different initial directory and users cannot access other users' private files. These features are discussed in this chapter.

The following section compares MS OS/2 *MULTIUSER* and MS OS/2 Release 1.21.

COMPARISON BETWEEN MS OS/2 RELEASE 1.21

AND CITRIX *MULTIUSER*

If you are already familiar with MS OS/2, you are already familiar with MS OS/2 *MULTIUSER*. The similarities, differences, and additional features are discussed below.

Similarities

Program Selector functions and command interpreter prompts are available for use. You can start multiple programs from the Program Selector, and you can switch between them and the Program Selector with a hotkey. MS OS/2 *MULTIUSER* runs MS OS/2 full screen text applications on the system console and on terminals. The MS OS/2 protect mode full screen text utilities and commands are the same as the MS OS/2 *MULTIUSER* commands and utilities.

Differences

There are differences in the way you use the MS OS/2 *MULTIUSER* system because you are working in a multiuser environment. Because of resource management, it is possible that a program you are running will not be able to use the maximum system limit of certain system resources. Depending on how your System Administrator has configured your security class, you may not have access to all the utilities available to System Administrators and Operators. In addition, you will not have access to all the files and directories that are on the fixed disk. Also, you may not have access to all of the devices on the system.

Because of the multiuser nature of MS OS/2 *MULTIUSER*, you probably will have to login in order to use it. In order to do this, you have been given a *username*, a *password*, and possibly a *groupname*.

Drive C will always be a High-Performance File System Partition (HPFS). If you have additional disks or partitions, they may be HPFS or FAT (file allocation table).

MS OS/2 *MULTIUSER* does not support the execution of MS DOS programs or programs and utilities that require the Presentation Manager execution environment. Programs cannot place the display in graphics mode. The system cannot be configured to run MS DOS from the fixed disk using the MS OS/2 Dual Boot feature.

Some terminals that MS OS/2 *MULTIUSER* supports have display limitations that may restrict you from performing certain functions in an application. For example, most terminals in use today are monochrome while most system consoles have significant color capabilities. This is not a limitation of MS OS/2 *MULTIUSER* but a limitation of some of the hardware you may be using.

Additional Features

MS OS/2 *MULTIUSER* is a multiuser system. Multiuser configuration utilities are available for a System Administrator and multiuser operation utilities are available for an Operator. This chapter discusses many of the multiuser features you may be exposed to during your use of MS OS/2 *MULTIUSER.*

Chapter 5 discusses some new multiuser utilities that you should familiarize yourself with because you are a user of MS OS/2 *MULTIUSER.*

SECURITY

Introduction to Security

Security provides a multitude of functions meant to ensure that you have a private and reliable working environment within the MS OS/2 *MULTIUSER* system. First, access to the system is limited, through use of a *loginname*, to only those users allowed. Then, by controlling and adjusting file system permissions, you are given a personal view of the system with private workspace. Also, through use of security classes, you are provided with a set of limitations to ensure that the integrity and privacy of the entire system remains intact.

Security During Login

Logging in to the system is like booting up a single-user system; it establishes a working environment for you through configuration and command files. Because your environment is different from other users, you are identified to the system by your *loginname*. This name has associated user configuration profiles that tell the system how to set the system up for you. Security provides login password verification to protect your resources from unauthorized access.

Resource Access Permissions

Once you have logged in, your data and applications are made available through the file system. The file system is global to all users on the system to allow sharing and maximum use of available space. File and directory permissions are maintained to provide you with your own personal view of the file system. A number of different permissions apply to files; typical permissions are Read (R), Write (W), Create (C), and Execute (X). In your working directory you would normally have all permissions, but you may only have RX permissions in a shared application directory.

Permissions are used to restrict the amount of filesystem data visible to you. Commands like DIR and other applications that view the filesystem directory tree will be restricted to access only the portion of the filesystem to which you have access permissions.

If you are not permitted to access a file, directory, or other resource (for example, a disk) the application or system usually reports an "access denied" error condition. In order to gain access, you need to have the System Administrator or owner of the resource authorize access for you.

Security Classes

Your user capabilities to perform system configuration and operational duties on MS OS/2 *MULTIUSER* are controlled by assignment to a security class. This is done when you are given a *loginname*. Users assigned to User class generally have complete control over their private filesystem resources and can look at, and sometimes change, system parameters that pertain directly to them. In contrast, a user assigned to Administrator class can manage system resources more globally, creating users, installing system-wide applications, and the like. The role of a System Administrator is described later in this chapter.

Security Utilities You May Need

You will need only a few basic commands to control your secure environment. First, you may like (or be required) to update your login *password*. You can do this at login time using password change parameters at the LOGIN prompt, or you can change your *password* using the PASSWORD command. Then you might need to give another user access to one of your files or directories. To do this you use the CONFIG ACCESS command sequence. This command provides fullscreen maneuvering through your files and allows you to give other users or groups of users access to your files.

Another utility you may need is RESERVE. This allows you to lock a device for exclusive use. This is useful when a single device, such as a diskette drive, is being shared among many users. Diskette drives A and B generally require RESERVE. Other devices may also require RESERVE. Your System Administrator will advise you when device reserve is required.

Use of these utilities is described in Chapter 5.

ESSENTIALS OF SHARED RESOURCES

Because MS OS/2 *MULTIUSER* is a multiuser system, the system resources can be divided among the users of the system. The system can be setup for both Minimum Guaranteed and Maximum Allowed Resource Limits. Not all resources have minimum limits; however, all resources do have maximum limits. You can use the QUERY LIMITS utility to determine if resource management is on and the current limits set for your *loginname.*

Minimum Guaranteed Resource Limit

Each user can be given a minimum guaranteed resource limit. This limit essentially reserves a certain amount of the resource for each user logged into the system. These limits are applied on a per user basis. For example, if you have a minimum limit of one megabyte (Mb) of virtual memory and you are only using 512K, other users will not be able to access the 512K that you are not using.

Resource limits are only in effect when you are logged on. If you login and not enough resources are available to satisfy your resource minimum limits, you will get an "insufficient resources" message. If this happens, contact your System Administrator.

Maximum Allowed Resource Limit

In order to prevent one user from taking all the available resources in the system, maximum allowed resource limits are imposed on a per user basis. If you exceed your maximum limit, you will receive an "out of resource" error message. If you receive this warning, contact your System Administrator.

It is important to note that you could run out of resources before you reach your maximum limit if other users have high minimum limits and the system is reserving the resources for them. If this occurs, contact your System Administrator.

ROLE OF THE SYSTEM ADMINISTRATOR

Your System Administrator is responsible for installing and configuring the MS OS/2 *MULTIUSER* system.

Using the resource management configuration functions, the System Administrator will decide what your minimum guaranteed resource limits are and what your maximum allowed resource limits are.

The System Administrator will decide what security class you reside in. This class determines whether some user functions are available to you. The System Administrator may also set up directories or files that may not be completely accessible to you.

The System Administrator will also assign you a *username* and initial *password*, and possibly a *groupname*. You will need this information to login. The System Administrator will also use this information for certain other security configuration tasks.

What To Do if You Have a Problem

If you have difficulties using the system and you cannot resolve them yourself, contact your System Administrator or a designated representative.

MULTIUSER DIRECTORY STRUCTURE

In order to support multiple users, there are differences in the handling of directories under MS OS/2 *MULTIUSER* compared to MS OS/2. These differences are discussed below.

In MS OS/2 *MULTIUSER*, the concept of a home directory for each user is introduced. Your home directory was created for you by the System Administrator when your MS OS/2 *MULTIUSER* *username* was defined. Each user's home directory is unique; it cannot be used as the home directory of any other user. The purposes of the home directory are to provide a place for your *username* specific startup file (*CONFIG.USR*) to reside and to act as a parent directory for your user specific files. For more information on startup files, see the section "Using Startup Files," in Chapter 6.

Another concept that is new in MS OS/2 *MULTIUSER* is that of a working directory for each user. Like your home directory, the working directory is defined for you by the System Administrator. In most cases, your working directory is the same as your home directory. However, this is not required. The purpose of the working directory is to define the directory that is your current directory when you first login to the system and to provide a place for the Program Selector startup file, *STARTUP.CMD*, to reside.

For example, consider the case of the *username* DAVIDH. DavidH's home directory is \USR\DAVIDH. This is where DavidH's startup files, if any, reside. If DavidH's working directory is the same as his home directory, \USR\DAVIDH is also DavidH's current directory immediately after he has logged on to the system.

A SAMPLE USER

A sample user is illustrated in the following sections.

User's Profile

User Name:	DAVIDH
Group Membership:	WRKGRP (also set as default)
Security Class:	USER
Password:	(yes)
First user program:	PSEL.EXE (Program Selector)
Working directory:	(defaulted to user home)

Logging In

At the login prompt, David would enter:

```
Login: DAVIDH
Password: _ _ _ _ _ _
```

This will log David into the system under the *loginname* of DAVIDH.WRKGRP

What Happens While Starting Up

Home Directory

David's home directory is \USR\DAVIDH with these files:

PSEL01.AIF
PSEL01.AII
STARTUP.CMD
CONFIG.USR

Startup Sequence

1. Data from *CONFIG.USR* is applied to David's environment.

2. PSEL starts and sets up the program selection list using the *PSEL01.AIF* and *PSEL01.AII* files.

3. PSEL then runs *CMD.EXE* for the *STARTUP.CMD* file. Because David's *STARTUP.CMD* file has an EXIT command, the Program Selector windows are displayed at completion.

User's Security Environment

David's security class is User; therefore, his access is limited to his home directory and system directories that contain user utilities. In addition, he has been given access to a TOOLS directory that has been set up by the System Administrator.

The following DIR commands issued by David at the command line (*CMD.EXE*) illustrate his secure working environment:

```
[C:\USR\DAVIDH] DIR

9-21-90   10:54a      <DIR>     0  .
9-21-90   10:54a      <DIR>     0  ..
9-21-90   10:54a       123      0  CONFIG.USR
9-21-90   10:54a      <DIR>     0  MYPROGRAMS
9-21-90   10:54a       333      0  SHELL11F.AIF
9-21-90   10:54a        86      0  SHELL11F.AII
9-21-90   10:54a        37      0  STARTUP.CMD
```

Note that all of the home files are accessible. In addition, David has a subdirectory with his private programs.

```
[C:\USR\DAVIDH]  DIR \

9-21-90   10:54a        <DIR>      0  OS2
9-21-90   10:54a        <DIR>      0  TOOLS
```

Although the system has many other subdirectories and files in the root of drive C, David can see only these. The OS2 directory has MS OS/2 *MULTIUSER* utilities. The TOOLS directory was set up by the System Administrator as accessible to David's group (WRKGRP).

```
[C:\USR\DAVIDH]  DIR ..

9-21-90   10:54a        <DIR>      0  .
9-21-90   10:54a        <DIR>      0  ..
9-21-90   10:54a        <DIR>      0  DAVIDH
```

This illustrates that David does not have access to any other user's home directory.

PRINTING

If a printer is attached to your system, several users may be trying to print at the same time. MS OS/2 *MULTIUSER* solves this problem by using printer queues. A printer queue is a temporary waiting place for data being sent to the printer. The system component that manages the print queues is called the spooler.

The spooler will choose print jobs from the queue and send them to the printer one at a time. The order in which they are printed is determined by the System Administrator.

PART 2

USING Citrix *MULTIUSER*

CHAPTER 3

USING THE COMMAND INTERPRETER

INTRODUCTION

This chapter explains how to start and use the command interpreter. Running commands like those used to manage files, directories, and setting up your command line are discussed. The concepts of environment, path, and redirection are covered.

RUNNING CMD

Introduction

The MS OS/2 *MULTIUSER* command interpreter, CMD, is a program that translates what you type at a prompt into commands that your computer can use. For example, when DAVIDH first starts a session running CMD, the prompt typically looks like the following:

```
[C:\USR\DAVIDH]
```

CMD is a program that runs in a session. CMD provides a command line interface, which means that CMD displays a prompt where you type commands to start other programs.

CMD contains a set of built-in commands that help you manage files and directories, create and run batch programs, and set system features. You can start other programs by using CMD, so it provides an alternative to the Program Selector.

This chapter describes how to start and quit the CMD program and how to use CMD commands. It also describes the KEYS command, which can be used to edit the CMD command line. For more information about utilities that can be run from the CMD program, see Chapter 4, "Using Citrix *MULTIUSER* Utilities."

This chapter also contains information about how the CMD program and CMD commands work in the High-Performance File System (HPFS). All examples of file and directory names in this chapter are for the HPFS file system.

Starting CMD

Program Selector

Move the selection bar over * OS/2 Command Prompt * in the Start Programs menu of the Program Selector and press **ENTER**. If you do not have an * OS/2 Command Prompt * selection in your Start Programs menu, you can add one the same way you add a program. Refer to the section on adding an application to the Start Programs List of the Program Selector.

First User Program

If the System Administrator configured your first user program as *CMD.EXE*, you will start with the CMD prompt when you login.

Hotkey

You can start a new session running CMD using the Session Create hotkey. This hotkey is typically **ALT+TAB** but may have been re-defined or disabled by your System Administrator.

Quitting CMD

At the command prompt, type EXIT and press **ENTER** to quit CMD.

NOTE: If the Administrator has configured you to start CMD as your first program, then if you **EXIT** from CMD and it is your last session, you are logged out.

Using Commands

To use a command in CMD, type the command on the command line and press **ENTER**. This section explains how to use the command line to start commands and also how you can use the features of the KEYS command to edit the command line.

Starting Commands

After you have started CMD, you are ready to use commands. At the CMD command prompt, type the name of the command, followed by any arguments, and press **ENTER**.

In addition to commands built into CMD, you can also start MS OS/2 *MULTIUSER* utilities and applications. To start other MS OS/2 *MULTIUSER* programs, see the manual that accompanies your program and "Running Programs," later in this chapter.

Editing the Command Line

When you are running CMD, you can perform editing functions on the command line by using the KEYS command. These editing functions will save you time by allowing you to call up commands you have already typed and using them again without retyping them. You start these functions by typing the following on the command line:

```
KEYS ON
```

Once you have typed the KEYS command, every command you enter on the command line is temporarily saved in a list in memory. To see the list of the commands that are currently in memory, type the following on the command line:

```
KEYS LIST
```

You can use any of the following keys and key combinations to perform the special editing functions that the KEYS command provides:

Key	Editing Function
ESC	Clears the current command line and returns the cursor to the position immediately following the command prompt.
HOME	Returns the cursor to the position immediately following the command prompt.
END	Places the cursor in the position immediately following the last character you typed (even if the last character you typed is a space).
INS	Turns the insert mode on and off. The cursor appears as a half box when you are in insert mode. As you insert characters, existing characters move to the right. At the beginning of each new command line, insert mode is turned off.
DEL	Deletes the character marked by the cursor. As you delete characters, any existing characters to the right of the cursor move to the left.
LEFT	Moves the cursor left one character.
RIGHT	Moves the cursor right one character.

Key	Editing Function
UP	Finds the previous command listed in memory and displays that command on the command line. When you reach the first command in the list, pressing **UP** displays the last command.
DOWN	Finds the next command listed in memory and displays that command on the command line. When you reach the last command in the list, pressing **DOWN** displays the first command.
BACKSPACE	Moves the cursor left one character and deletes that character. Any characters to the right of the cursor move left to fill the deleted character's position.
CTRL+LEFT	Moves the cursor left to the first character of a word. If the cursor is already positioned on the first character of a word or in the space between words, the cursor moves to the first character of the word to the left.
CTRL+RIGHT	Moves the cursor right to the first character of the next word.
CTRL+END	Deletes any characters from the current cursor position to the end of the command line.
CTRL+HOME	Deletes any characters from the beginning of the command line up to (but not including) the current cursor position.

Key	Editing Function
ENTER	Sends the information on the command line to the command interpreter and adds it to the list of commands in memory. Pressing **ENTER** also turns off insert mode.

For example, suppose you have specified KEYS=ON and when you type KEYS LIST to see the list of commands in memory, the following list is displayed:

```
1: CD ACCOUNTS
2: DIR SALES.FEB
3: COPY A:\SALES.FEB
4: CD ..
5: PATH
6: TYPE SALES.FEB
7: KEYS LIST
```

Now suppose you want to use any of the keys' editing functions. Press the **UP** key to bring the last command in the list to the command line. Press **UP** again to display the previous command. As you continue to press **UP**, the KEYS command takes you through the entire list and returns to the last command after reaching the beginning of the list.

The KEYS command continues to add commands to the list until you type KEYS OFF on the command line or exit the MS OS/2 *MULTIUSER* session in which it was started. To find out whether the KEYS command is on or off, type KEYS and press **ENTER**.

Your System Administrator can have MS OS/2 *MULTIUSER* start the KEYS command every time you login.

GETTING HELP INFORMATION

When you use CMD commands, error messages may appear on your screen. You can use the HELP program to get an explanation of any error message. Each message consists of a number (for example, SYS1002) and a brief description of the error condition. For a more detailed explanation of the error condition, type HELP followed by the message number. For example, suppose that when you try to get a directory list in wide format, you accidentally type DIR /Z instead of DIR /W. You receive the following error message, "SYS1003: The syntax of the command is incorrect." To see a detailed explanation of this error message, type the following:

```
HELP SYS1003
```

The following explanation is provided:

```
SYS1003: The syntax of the command is incorrect.

EXPLANATION: One of the following occurred:

1. An incorrect parameter was specified.
2. A required parameter is missing.
3. Too many parameters were entered.
4. The parameters were entered in the wrong order.

ACTION: Check the syntax of the command. Then
retry the command.
```

SETTING UP YOUR COMMAND LINE

When you first set up your command line, you may want to do such things as customize the MS OS/2 *MULTIUSER* command prompt or display the date and the time of day. The CMD program contains several built-in commands that help you perform these types of tasks. This section describes some of the commands you might use when setting up your system. The commands are as follows:

Command	Purpose
CLS	Clears your screen.
DATE	Displays the system date.
TIME	Displays the system time.
PROMPT	Changes the CMD prompt.
MODE	Changes the display mode.

These commands are described in the following sections.

Clearing the Screen

The CLS command clears your terminal screen. Once you clear the screen, you cannot retrieve the information. To use this command, type the following:

```
CLS
```

Your screen clears and the MS OS/2 *MULTIUSER* command prompt appears in the upper left corner of the screen.

Displaying the Date

You can display the system date by using the DATE command. MS OS/2 *MULTIUSER* uses this date to update the directory list whenever you create or change a file or directory. The date set by the System Administrator applies to all sessions. To use the DATE command, type the following:

```
DATE
```

You'll see a message similar to this:

```
The current date is Fri 3-24-1989
Enter the new date: (mm-dd-yy)
```

Press **ENTER** to return to the command line.

NOTE: Only an Administrator or Operator class user can change the date. Any changes attempted by you are rejected.

Displaying the Time

You can display the system time with the TIME command. MS OS/2 *MULTIUSER* uses this time to update the directory list whenever you create or change a file or directory. The time set by the System Administrator applies to all sessions. To use the TIME command, type the following:

```
TIME
```

You'll see a message similar to this:

```
The current time is: 10:21:39.03
Enter the new time: (hh:mm:ss)
```

Press **ENTER** to return to the command line.

NOTE: Only an Administrator or Operator class user can change the time. Any changes attempted by you are rejected.

Changing the CMD Prompt

You can change the way your prompt looks by using the PROMPT command. PROMPT recognizes several character combinations, all of which begin with a dollar sign (**$**). Each character combination displays a different prompt. The PROMPT command affects only the current session. To change the prompt, type PROMPT followed by one or more character combinations. For example, to change the prompt to an equal sign followed by a greater-than sign, type the following:

```
PROMPT $Q$G
```

Now your prompt looks like this:

```
=>
```

You can use any of the following character combinations to create your prompt:

Characters	Prompt
$$	Dollar sign ($)
$t	Current time
$d	Current date
$p	Current directory on the current drive
$v	Version number
$n	Current drive letter
$g	Greater-than sign (>)
$l	Less-than sign (<)
$b	Pipe symbol (|)
$_	New line (the equivalent of pressing ENTER)
$e	ANSI escape character
$q	Equal sign (=)
$h	Backspace (to erase a character in the prompt)
$i	Help line
$c	Left parenthesis [( ]
$f	Right parenthesis [ )]
$a	Ampersand (&)
$o	Current hostname
$u	Current username

You can also change your prompt to any string of characters or combination of characters and symbols. For more information on how to change your prompt to a string of characters, see the *Citrix MULTIUSER Command Reference.* If you type PROMPT by itself, the prompt changes to the default system prompt.

Setting the Display Mode

To set the current display mode, use the MODE command. Several options can be used with this command. Note that not all terminals can support all modes. Specific information on the modes supported by a given terminal is covered in Appendix A, "Terminals Used With Citrix *MULTIUSER.*"

The general format of the MODE command is:

```
MODE <type>,<lines>
```

<type>

Display type (40, 80, BW40, BW80, CO40, CO80, MONO)
40 = 40-column mode
80 = 80-column mode
BW40 = 40-column mode (color adapter, color disabled)
BW80 = 80-column mode (color adapter, color disabled)
CO40 = 40-column color mode (color adapter)
CO80 = 80-column color mode (color adapter)
MONO = 80-column monochrome mode (monochrome adapter)

Most terminals will support only 80-column modes. All terminals support modes MONO and CO80.

<lines>

Number of lines displayed (25, 43, 50, 60)

If the number of lines is not specified, MODE uses the current number of lines.

Most terminals support only 25-line modes; some also support other numbers of displayed lines (usually 43-line modes). Check Appendix A, "Terminals Used With Citrix *MULTIUSER*" for terminal specific details. Consult with the System Administrator if your terminal is not listed in Appendix A and you need this information.

Some sample MODE commands are:

To set the display into 80-column monochrome mode, type the following:

```
MODE MONO
```

Note that the number of lines displayed does not change.

NOTE: The system console and all terminals support this mode.

To set the display into 25 X 80 color mode, type the following:

```
MODE CO80,25
```

NOTE: The system console and all terminals support this mode.

To set the display into 43 X 80 color mode, type the following:

```
MODE CO80,43
```

NOTE: The system console and some terminals support this mode.

USING HIGH-PERFORMANCE FILE SYSTEM (HPFS)

Introduction

A feature of MS OS/2 *MULTIUSER* is the capability for multiple file systems to exist under one operating system. The original file allocation table (FAT) file system is still present, but you can install additional file systems as well. In MS OS/2 *MULTIUSER*, an additional file system is referred to as an installable file system (IFS).

One installable file system that is shipped with MS OS/2 *MULTIUSER* and must be used on the boot partition is the High-Performance File System (HPFS). HPFS completes the tasks of writing to and reading from your fixed disk much faster than the FAT file system can. HPFS also allows you to use up to 254 characters, including some characters not allowed in the FAT file system, when naming files and directories; this lets you create more descriptive names.

HPFS also supports the MS OS/2 *MULTIUSER* security attributes, allowing the System Administrator to control access permissions to files and directories.

Naming rules and the interaction of HPFS filenames with MS OS/2 *MULTIUSER* are described in the following sections:

Naming HPFS Files and Directories

HPFS file and directory names may include several features that are not available in the FAT file system:

- File and directory names can be up to 254 characters long. (The FAT file system has a limit of 8-character filenames and 3-character filename extensions.) Paths and filenames together can be up to 259 characters long.

- Blank spaces and periods (.) can occur anywhere in the file or directory name. However, blanks and/or periods that occur at the end of a name are not treated by MS OS/2 *MULTIUSER* as a significant part of the name. For example, the filenames "*xyz*", "*xyz.*", "*xyz.* ", and "*xyz* ." are stored by MS OS/2 *MULTIUSER* as "*xyz*".

- The following characters can be used in naming HPFS files and directories:

  , + = [ ] ;

  Note that the following characters are not currently allowed in any file system with MS OS/2 *MULTIUSER*:

  < > : " / \ | * ?

- You can use uppercase, lowercase, or mixed case when naming HPFS files and directories; the name is displayed in the directory list just as you typed it. However, MS OS/2 *MULTIUSER* ignores case in comparing file and directory names. For example, "*Taxfile*", "*TAXFILE*", and "*taxfile*" are the same filename to MS OS/2 *MULTIUSER*; only one can exist in a given directory.

Using HPFS File and Directory Names With Commands

You must enclose any blanks or special characters in double quotation marks when you type an HPFS filename or directory name on the command line so that MS OS/2 *MULTIUSER* will recognize that they are part of the name. You can put the double quotation marks around the entire name or around just the blanks and special characters. For example, to copy a file named "*My Tax File, 1988.txt*" to a directory on drive C named "Current Taxes", you can type either of the following:

```
COPY "My Tax File, 1988.txt"  C:"Current Taxes"
COPY My" "Tax" "File", "1988.txt  C:Current" "Taxes
```

You must also use double quotation marks any time a file or directory that uses the HPFS naming features is listed in batch programs.

Transferring HPFS Files and Directories

Because not all file systems support the use of HPFS naming features, you must be careful when copying and moving files between file systems. In general, HPFS file and directory names that use HPFS naming features cannot be transferred from HPFS to the FAT file system. (If you use the COPY or MOVE command at the command prompt, you must rename the file; the HPFS filename is lost.)

NOTE: In HPFS, you can change the case of a file or directory name (for example, from uppercase to lowercase) by using the MOVE or the RENAME command and giving the file or directory a new name. You can use the MOVE or RENAME command at the command prompt; when you type the new name, use the case you want the name to have. For example, to change *MYTAXFILE.TXT* to lowercase from the command line, you would type the following:

```
MOVE MYTAXFILE.TXT mytaxfile.txt
```

Using Wildcard Characters in HPFS

You can use the asterisk (*) and question mark (?) wildcard characters in HPFS. Wildcard characters take the place of other characters when you specify filenames and directories in commands and utilities, and are especially useful when you work with groups of files.

Generally, the asterisk matches zero or more characters in a filename. For example, suppose you have the following files:

M
MAY
MAY.TXT
MAY.TXT.BAK

If you type MAY* when you are using the CHDIR, DEL, DIR, or RMDIR commands in HPFS, the files *MAY*, *MAY.TXT*, and *MAY.TXT.BAK* are matched. If you type *. instead, the files *M* and *MAY* are matched.

When you use the COPY, MOVE, or RENAME commands in HPFS, the asterisk works a little differently. With each of these commands, you specify source and destination filenames. When you use the asterisk in the source filename, it works as previously described. However, when you use the asterisk in the destination filename, the system copies the characters that follow the asterisk to the destination filename. For example, if you type COPY *.C *.BAK, the COPY command first finds all files that end with *.C*. Then it makes a copy of each file, using as the name of the new file the characters that precede the *.C* and adding the new *.BAK* extension.

The question mark matches any single character in a filename except a period (.). For example, *JUNE?* matches *JUNE*, *JUNE1*, and *JUNE2*, but does not match *JUNE3.TXT* or *JUNE.TXT*.

Wildcard characters are valid only in the filename that follows the last backslash (\) of a path.

Security and HPFS

It is important to note that file and directory security can be done on the HPFS file system, but not on the FAT file system. Thus, when files are moved from HPFS to FAT, the security attributes will no longer apply.

MANAGING FILES, DIRECTORIES, AND DRIVES

MS OS/2 *MULTIUSER* provides several commands that help you manage files, directories, and drives. These commands, with their alternate forms shown in parentheses, are as follows:

Command	Purpose
DIR	Displays the contents of a directory.
MKDIR (MD)	Makes a new directory.
CHDIR (CD)	Switches to a different directory.
RMDIR (RD)	Removes a directory.
TYPE	Displays the contents of a file.
COPY	Copies a file.
MOVE	Moves a file or directory.
DEL (ERASE)	Deletes a file.
RENAME (REN)	Renames a file or directory.
VOL	Displays the volume identification of the current drive.

Unless you specify otherwise, all commands that you type at the CMD prompt work with the contents of the current directory. Many of these commands accept options, which are letters preceded by a slash (/), that modify how a command works. While most command options are described in this chapter, see the *Citrix MULTIUSER Command Reference* for a complete list and description of the options for each command.

Displaying the Directory Information

To display a list of the contents of a directory, use the DIR command. A directory list consists of individual directory entries, each of which describes one file or subdirectory (a directory created within another directory). You can see only the files and directories you have access to.

You can display the directory list for your current directory, for a specific directory or file, for more than one directory, or for a group of files. In addition, the DIR command can display the list in a multiple-column format or a page-by-page format. You can also choose the order in which the directory information is displayed. The various options are described in the following sections.

Displaying the Contents of the Current Directory

To display the contents of the current directory, you type the DIR command without any arguments.

For example, suppose you are working in a directory called \USR\DAVIDH\FINANCE. To display the directory entries, type the following:

```
DIR
```

The CMD command interpreter displays a list similar to the following:

```
The volume label in drive C is DAVID.
The Volume Serial Number is E392:1C15
Directory of C:\USR\DAVIDH\FINANCE

 1-30-89   9:10p      <DIR>     0  .
 1-30-89   9:10p      <DIR>     0  ..
 1-30-89   9:11a      <DIR>     0  BUSINESS
 1-30-89   9:12a      <DIR>     0  PERSONAL
10-15-89   8:47a      <DIR>     0  TAXES
 4-20-89  10:31a       452      0  1QUARTER.SUM
 7-27-89   9:05a       590      0  2QUARTER.SUM
10-08-89   3:54p       524      0  3QUARTER.SUM
11-08-89   2:15p      1168      0  REPORT.TXT
     9 File(s)   1880064 bytes free
```

Here is what each directory entry contains:

- The volume label and serial number of your current drive.
- The name of the current drive and directory.
- The date that the file or directory was created or last modified.
- The time that the file or directory was created or last modified.
- The size of the file (in bytes).
- Size of the extended attributes (EAs).
- The filename and filename extension (if any) or the directory name.

Directories have <DIR> listed after their names. At the end of the list, DIR displays the number of files in the directory and the number of bytes that are free on the disk. It is a good idea to check your available disk space regularly, especially if your disk space is limited.

If you are viewing a directory list for a directory other than the root directory by using the file allocation table (FAT) file system, you'll see periods (. and ..) listed as directory entries. These characters are a shorthand notation for the current (.) and parent (..) directories. The current directory is the one you are now working in, and the parent directory is one level above the current directory.

In HPFS, you'll see these entries in the root directory as well. Because the root directory has no parent directory, the parent (..) notation listed there has no meaning.

You can use this notation in any CMD command or utility to specify the current or parent directory. For example, to see the directory list for a parent directory, type the following:

```
DIR ..
```

You can also display directory entries for the current directory on a different drive. For example, to see what is in the current directory on drive C, type the following:

```
DIR C:
```

A list of the files and directories for drive C appears on your screen. You may want to see only a directory list of filenames instead of the complete directory entries described previously. For a list of the filenames on drive C, type the following:

```
DIR C: /F
```

When you use the /F option, the DIR command also displays the complete directory path of the files.

Viewing a Directory List for a Specific File or Directory

The DIR command lets you view any directory list without changing your current directory.

To do this, you must tell CMD the path of the directory you want to view. You can specify the directory's full path from the root directory or you can specify a relative path from your current directory. Although you can always type the full path of a directory, it is not required. For example, suppose your business travel schedules are stored in the TRAVEL subdirectory of the FINANCE\BUSINESS directory. To view the files in your TRAVEL subdirectory from the root directory, type the following:

```
DIR \USR\DAVIDH\FINANCE\BUSINESS\TRAVEL
```

However, if you are in the FINANCE directory, you only need to type the path from the current directory to the TRAVEL subdirectory:

```
DIR BUSINESS\TRAVEL
```

If you prefer to see personal travel schedules that are stored in the TRAVEL subdirectory of the FINANCE\PERSONAL directory on drive C, type the following:

```
DIR C:\USR\DAVIDH\FINANCE\PERSONAL\TRAVEL
```

NOTE: If you will be doing much work in another directory, you can also use the CHDIR command to move to that directory before using the DIR command. The CHDIR command is described in detail in "Changing to Another Directory" later in this chapter.

Displaying the Contents of Multiple Directories

You can display directory lists for more than one directory by typing the name of each directory after the DIR command. For example, suppose that you are planning to purchase new desks and lamps for your employees. Although you have price lists for many items in your current directory, you would like to see the lists that pertain only to desks and lamps. To see the directory lists for the DESKS directory, which is on your current drive, and the directory list for the LAMPS directory, which is on drive C, type the following:

```
DIR DESKS C:\USR\DAVIDH\LAMPS
```

You will first see the list for DESKS, followed by the list for LAMPS:

```
The volume label in drive C is INVENTORY.
The Volume Serial Number is 0A73:331B
Directory of C:\USR\DAVIDH\DESKS

 6-23-89   2:58p   <DIR> 0  .
 6-23-89   2:58p   <DIR> 0  ..
 6-09-89  12:35p    899  0  STUDY.DSK
 2-02-89   1:41p   3821  0  TOOLS.DSK
 3-16-89   8:40a     16  0  WORK.DSK
     5 File(s)   1646592 bytes free

The volume label in drive C is INVENTORY.
The Volume Serial Number is 0A73:331B
Directory of C:\USR\DAVIDH\LAMPS

 4-28-89   6:30p   <DIR> 0  .
 4-28-89   6:30p   <DIR> 0  ..
 1-09-89   1:02p    240  0  DESK.LMP
 4-16-89   3:54p   1234  0  TABLE.LMP
 2-10-89   8:59a     80  0  SPOT.LMP
     5 File(s)   512 bytes free
```

If the DIR command cannot find one of the specified directories, it displays an error message telling you which directory could not be found and then displays the other directory.

Displaying a Directory List for a Group of Files

You might find that you want to view a directory list for a specific group of files. For example, to view a directory list in your current directory for all of the files that have the filename extension .DOC, type the following:

```
DIR *.DOC
```

The asterisk (*) is a wildcard character, which has a special meaning to CMD. The CMD program replaces the asterisk with any alphanumeric character or string of characters. In the previous example, the files *REPORT.DOC*, *REPLY.DOC*, and *LETTER.DOC* are all displayed.

Wildcard characters can be used with the DIR command to specify other groups of files. For example, to view all of the files that begin with the letter "R," type the following:

```
DIR R*
```

This displays the files *REPORT.DOC* and *REPLY.DOC*, but not *LETTER.DOC*.

To view all files beginning with the characters FIN, regardless of their filename extensions, type the following:

```
DIR FIN*
```

The files *FINANCE.TXT*, *FIND.TXT*, and *FIN024.DAT* are all included in the list. In addition to the asterisk, the question mark (?) can also be used as a wildcard character. A question mark in a filename or filename extension means that any one character or no character at all may occupy that position. For example, to display a directory list for trade shows that you have attended, you could type the following:

```
DIR SHOW?.*
```

You would see a list that contains *SHOW.PDX*, *SHOW1.SEA*, *SHOWN.SAF*, and *SHOW3.LAX*; it would, however, not include *SHOW23.SEA* (since the question mark stands for only one character).

Displaying a Directory List in Wide Format

If you have a large directory with many files, you might not be able to see all of the directory lists on one screen. One way to condense this list is to display it in wide (/W) format.

Suppose that your ACCOUNTS directory contains files for every account you have. You want to see the names of all the accounts, but you know that it is a long list of names. To display the directory list in wide format, type the /W option on the command line:

```
DIR ACCOUNTS /W
```

You see a multiple-column list of the filenames in the directory and they appear without the file size or date/time information.

Viewing a Directory List One Page at a Time

Just as you read a book one page at a time, you can view a directory list one screenful at a time. To do this, use the /P option:

```
DIR ACCOUNTS /P
```

The first part of the directory list is displayed, then the message "Press any key when ready . . ." appears at the bottom of the screen when the screen is full. After you press a key, the second screenful of the list appears. This continues until all of the directory entries are displayed or until you press **CTRL+C** to stop the operation.

You can also use the /W and /P options together. This option combination displays directory lists in wide format, one screenful at a time.

Creating a Directory

You can create a directory by using the MKDIR (MD) command. You can create a directory under your current directory, or you can create a directory in a specified drive or location.

You may not have access to the root directory. You can create additional directories that branch out from a directory into a multilevel directory structure. Directories created within other directories are sometimes referred to as subdirectories.

Creating a Directory in the Current Directory

To make a subdirectory in your current directory, type MKDIR or MD followed by the name of the new directory. For example, to create a new directory named INVOICE under the FINANCE directory, go to the FINANCE directory and type the following:

```
MD INVOICE
```

If you now type DIR, you'll see INVOICE listed as a directory:

```
 6-01-89  4:44p <DIR> 0  INVOICE
```

Creating a Directory in a Specified Location

You can also make a new directory anywhere you choose. This means that even if you are in the INVOICE directory on drive C, you can make a new directory named LETTERS in the home directory of DAVIDH on drive C. Just specify the drive and path of the new directory after the MKDIR command, as follows:

```
MD  C:\USR\DAVIDH\LETTERS
```

This creates a new directory on drive C named LETTERS.

Creating Multiple Directories

To save time, you can create more than one directory at once. For example, to create the SUPPLIES directory within your current directory and the EXPENSES directory within the current directory on drive C, type the following:

```
MD SUPPLIES C:EXPENSES
```

This creates two directories: one on your current drive and one on drive C. If the MKDIR command cannot create both directories, it displays an error message telling you which directory cannot be created, but continues to create the other directory.

Changing to Another Directory

If you want to work in a different directory, you can use the CHDIR (CD) command to change directories. To change to a subdirectory within the current directory, type CHDIR or CD followed by the name of the subdirectory. For example, to change from the FINANCE directory to the INVOICE subdirectory, type the following:

```
CD INVOICE
```

To change to a directory somewhere else on the directory tree, you must specify the path of the directory.

Moving to the Parent or Home Directory

Because of file system security, you may not have access to the root directory. You have seen how the DIR command uses periods to list the current (.) and parent (..) directories in a directory list. You can easily change to the current directory's parent directory by typing the following:

```
CD ..
```

Each time you type this command, you move one level closer to the root directory. For example, suppose you are in the MEMOS directory, the full path of which is \USR \DAVIDH\FINANCE\LETTERS\MEMOS. To return to the \USR\DAVIDH\FINANCE directory, type the following:

```
CD ..\..
```

This takes you out of the MEMOS subdirectory and moves you to the FINANCE directory. You can also combine the periods (..) with directory names to specify a directory. For example, suppose you are in the \USR\DAVIDH\LETTERS \WORK directory and you want to change to the \USR

\DAVIDH\LETTERS\FAMILY directory. You could move up one level by typing CD .. and then type CD FAMILY to move down to the right directory. However, an easier way is to combine the two operations in one step, as follows:

```
CD ..\FAMILY
```

To quickly change to DAVIDH's home root directory, regardless of where you are in the directory structure, type the following:

```
CD \USR\DAVIDH
```

Removing a Directory

The RMDIR (RD) command removes a directory.

Before you can remove a directory, you must delete all the files in the directory. You must also change to a directory other than the one you want to delete, since the CMD program will not let you delete a directory that you are currently in. For information on deleting, see "Deleting a File" later in this chapter.

NOTE: You cannot remove the root directory and you cannot remove directories that are being used in other sessions. You also cannot remove your home directory.

Removing One Directory

To remove one directory, you must be sure that the directory is empty and that you are in a different directory before you type RMDIR or RD followed by the name of the directory you want to delete. For example, to delete the directory LETTERS, do the following:

1. If you are in the \USR\DAVIDH\LETTERS directory, type CD .. to switch to the parent directory.

2. Type DEL LETTERS* to delete all of the files in the directory. A message will appear, asking whether you are sure that you want to delete all files. Type **Y** to delete all files in the directory.

3. Type RD LETTERS to remove the directory.

Removing Multiple Directories

To remove more than one directory, type the RD command, then specify each directory you want to remove, as follows:

```
RD  LETTERS  TRAVEL
```

Make sure that each directory you want to remove is empty. If the RD command cannot remove both directories, it displays an error message telling you which directory could not be removed and continues to remove the other directory.

Displaying the Contents of a File

To display the contents of a data file, use the TYPE command. For example, suppose that you have saved a letter in a file called *MEMOJAN.FIL.* To view the contents of *MEMOJAN.FIL*, type the following:

```
TYPE MEMOJAN.FIL
```

The contents of the file scroll on your screen:

```
TO:       John Howard
FROM:     Elizabeth Johnson
SUBJECT:  Questionnaires
DATE:     January 12, 1989

The twenty questionnaires you sent to our
department have been completed and are enclosed in
the accompanying envelope.

Thank you for encouraging us to participate in
your survey. We look forward to learning the
results!
```

If the file is large, you may want to temporarily stop the text from scrolling on your screen. To do this, press **CTRL+S**. To resume scrolling, press any key.

NOTE: If you have a large file that you want to view, you might want to use the MORE utility instead of the TYPE command. The MORE utility displays the contents of a file one screenful at a time. For information about the MORE utility, see Chapter 4, "Using Citrix *MULTIUSER* Utilities."

You can display more than one file at a time by adding additional filenames after the TYPE command. For example, to display the files *MEMOJAN.FIL* and *MEMOFEB.FIL*, type the following:

```
TYPE MEMOJAN.FIL MEMOFEB.FIL
```

The contents of *MEMOJAN.FIL* appear first, followed by the contents of *MEMOFEB.FIL*. If the TYPE command cannot find one of the specified files, it displays an error message telling you which file could not be found and continues to display the other file.

You could also use wildcard characters to display several files. For example, to display the contents of all the files with the extension *.FIL*, one after the other, type the following:

```
TYPE *.FIL
```

You can also use redirection symbols with the TYPE command. Redirection is a feature that lets you take the output from a command and send it to a file instead of to the screen. For information about redirection symbols, see "Redirecting Input, Output, and Error Messages," later in this chapter.

Copying a File

You can copy files from one location to another by using the COPY command. In addition, COPY can combine files and create files. Just specify a source file that will be copied and a destination file that will receive the copy. You can copy ASCII as well as binary files; the default setting is binary.

Copying One File to Another

To copy the contents of one file to another file, type COPY followed by the names of the source file and the destination file. For example, to copy the contents of the file *OLD.TXT* to the file *NEW.TXT*, type the following:

```
COPY  OLD.TXT  NEW.TXT
```

If *NEW.TXT* does not currently exist, it is automatically created, then *OLD.TXT* is copied to it. If *NEW.TXT* does exist, it is replaced by the contents of *OLD.TXT*. Be careful when you copy the contents of a file to an existing file, because the contents of the existing file will become permanently lost.

If you do not specify a directory path when using the COPY command, MS OS/2 *MULTIUSER* copies files in the current directory. If you want to specify files in other directories, make sure that you include their directory paths. For example, to copy the contents of *OLD.TXT* in the current directory to *NEW.TXT* in drive C, type the following:

```
COPY  OLD.TXT  C:NEW.TXT
```

To copy the contents of a file on drive C called *INVOICE.DOC* to your current directory and use the same filename, type the following:

```
COPY  C:INVOICE.DOC
```

Copying a Group of Files

You can use wildcard characters to copy a group of files to another group of files. For example, to copy the contents of all of the files in the current directory that have the filename extension *.EXE* to an identical set of files in drive C, type the following:

```
COPY *.EXE C:
```

You can also copy a group of files to a specific directory. For example, to copy the contents of all the *.TXT* files to the LETTERS subdirectory in the FINANCE directory, type the following:

```
COPY *.TXT \USR\DAVIDH\FINANCE\LETTERS
```

There may be times when you will want to copy an entire directory into another directory. For example, to copy all of the files in the current directory into the REPORTS directory in drive C, type the following:

```
COPY * C:\USR\DAVIDH\REPORTS
```

Be sure the destination directory exists; if it does not, MS OS/2 *MULTIUSER* will create a file called REPORTS in the home directory of DAVIDH on drive C, then copy the contents of all the files into that one file.

NOTE: If you want to copy the contents of an entire floppy disk onto another floppy disk, use the DISKCOPY utility, which is described in the *Citrix MULTIUSER Command Reference.*

Appending a File to Another File

In addition to copying files, the COPY command can append one or more files to an existing file or combine them into a new file. To do this, list any number of files as arguments to the COPY command. The files to be combined should be separated by plus signs (+), and you can specify a destination file that the combined files will be copied to. (The original files will still exist, unless you have made one of them the destination file.) For example, suppose that you want to combine two files named *JAN.FIL* and *FEB.FIL*. To add the contents of *FEB.FIL* to *JAN.FIL*, type the following:

```
COPY JAN.FIL+FEB.FIL
```

When the operation is completed, CMD displays the following message: "1 file(s) copied." The plus sign (+) between the files means that the contents of the *FEB.FIL* file will be added to the *JAN.FIL* file. (*FEB.FIL* will still exist in its original form.)

You can combine several files and copy them to another file. For example, if you want to add the contents of the files *JAN.FIL*, *FEB.FIL*, and *MAR.FIL* to a new file called *WINTER.FIL*, type the following:

```
COPY JAN.FIL+FEB.FIL+MAR.FIL WINTER.FIL
```

If you specify a destination file when you append files, the destination file is created and given the current date and time. If you omit a destination file, MS OS/2 *MULTIUSER* combines the files and stores them under the name of the file specified first.

You can also combine several files into one file by using wildcard characters. The following command takes all files with the *.TXT* extension and combines them into one file named *COMBIN.FIL*:

```
COPY *.TXT COMBIN.FIL
```

In the following example, each file that has the extension *.TXT* is combined with its corresponding *.REF* file. The result in each case is a file with the same filename, but with the extension *.FIL*. For example, *VIDEO.TXT* would be combined with *VIDEO.REF* to form *VIDEO.FIL*, *AUDIO.TXT* would be combined with *AUDIO.REF* to form *AUDIO.FIL*, and so on, if you type the following:

```
COPY *.TXT+*.REF *.FIL
```

If *.TXT* files do not correspond exactly with *.REF* files, existing files are copied singly to the *.FIL* files.

The following COPY command combines all of the files with the extension *.TXT* and all of the files with the extension *.REF* into one file named *COMBIN.FIL*:

```
COPY *.TXT+*.REF COMBIN.FIL
```

Creating a File

In the previous examples, you have seen how to copy files to other files. But the COPY command can also copy to a file what you type on the keyboard. You need only to specify

CON (for console) after the COPY command. In MS OS/2 *MULTIUSER*, the keyboard is referred to as CON. For example, to create the file *MESSAGE.TXT*, type the following:

```
COPY CON MESSAGE.TXT
```

Then press **ENTER** and type the text of the message. For example, you could type the following:

I have just received your shipment of invoices. Thanks for your prompt response.

K. P. Smith

To end the note and save it in the file, press **CTRL+Z** followed by **ENTER**. You will see a message saying "1 file(s) copied", then the command prompt again.

If you use the DIR command to view the directory entries, you will see that the file *MESSAGE.TXT* now appears in your directory.

NOTE: The COPY command has four additional options. The /V option verifies a copy operation, the /A option copies ASCII files, the /B option copies binary files, and the /F option indicates that extended attributes should not be discarded if the destination file system does not support them. For more information on how to use these options, see the *Citrix MULTIUSER Command Reference.*

Preserving Extended Attributes

In MS OS/2 *MULTIUSER*, some applications attach additional information (such as author, application type, and file history) to your files and directories. This type of information is called extended attributes and is used by other applications, the file system, or the operating system itself. For example, an application such as Microsoft Excel may look at a file's extended attributes to see if the file is a Microsoft Excel chart.

Take care when using the COPY command to preserve any extended attributes. When you are simply making a copy of a file, the COPY command preserves all extended attributes. However, when you are appending one or more files to an existing file or combining files into a new file, only the extended attributes of the first file specified are preserved.

It is possible for a file to have zero length and yet still have extended attributes. You can use the COPY command to copy a file with zero length; the COPY command copies and preserves these extended attributes.

If you are copying a file that has extended attributes to a file system that does not support extended attributes, you may want to use the /F option with the COPY command. For more information about using the COPY command on files with extended attributes, see the *Citrix MULTIUSER Command Reference.*

You may want to use the EAUTIL utility before copying files that have extended attributes. By using the EAUTIL utility, you can split files and their extended attributes into separate files, copy the files, and then join the files together again. For information about the EAUTIL utility, see the *Citrix MULTIUSER Command Reference.*

You can see whether your files have extended attributes on a FAT file system by using the DIR command with the /N option. If a file has extended attributes, the size of the attributes is displayed in the column before the filename. For example, to see if the files in the LIBRARY directory have extended attributes, type the following:

```
DIR /N
```

You'll see a directory list similar to the one that follows. The number in the column before the filename tells you the size of its extended attributes (a zero tells you there are no extended attributes associated with that file or directory).

```
The volume label in drive C is DAVIDH.
The Volume Serial Number is E392:1C15
Directory of C:\USR\DAVIDH\LIBRARY

8-02-89      11:02a     <DIR>    0  .
8-02-89      11:02a     <DIR>    0  ..
8-22-89       2:41p    119202    4  CATALOG.TXT
8-22-89       3:36p     65288    4  BOOKS.TXT
   4 File(s)      1646592 bytes free
```

Moving Files and Directories

You can move files and directories from one location to another on the same drive by using the MOVE command. You may want to move files and directories when organizing them on your fixed disk.

Moving a File

To move the file *SCHEDULE.TXT* from the \USR\DAVIDH\MONTHLY directory on drive C to the HISTORY directory on that same drive, type the following (on the same line):

```
MOVE C:\USR\DAVIDH\MONTHLY\SCHEDULE.TXT
\HISTORY
```

In this example, the file is simply moved; its name stays the same. You can, however, move a file to another directory and give it a new name at the same time. Just specify the new filename after the directory name.

You can use wildcard characters with the MOVE command to move multiple files or directories, just as you can use them with the COPY command.

Moving a Directory

You can also move a directory and its entire contents to another location on the same drive. Suppose you want to move the \USR\DAVIDH\UTILITY directory on drive C into the directory OS12. To do this, type the following:

```
MOVE C:\USR\DAVIDH\UTILITY  \OS12
```

In this example, the \USR\DAVIDH\UTILITY directory becomes a subdirectory of the OS12 directory.

Deleting a File

Just as you may want to make copies of files or move files around, you also may want to remove files. When you want to delete a file from a disk, you can use the DEL or ERASE command.

Deleting One File

You can delete one file from a directory by typing DEL or ERASE followed by the name of the file. For example, to delete the file *OLD.TXT* from the home directory in drive C, type the following:

```
DEL C:\USR\DAVIDH\OLD.TXT
```

Use the DIR command to verify that the file has been deleted. It is a good idea to go through your directories periodically and delete unnecessary files. Unnecessary files take up valuable disk space.

Deleting Multiple Files

You can delete more than one file at a time by typing more than one filename on the DEL command line. For example, to delete the file *OLD.TXT* from the root directory in drive A, the file *OLD2.TXT* from the root directory of drive B, and the file *OLD3.TXT* from the home directory of drive C, type the following (on the same line):

```
DEL A:\OLD.TXT B:\OLD2.TXT
C:\USR\DAVIDH\OLD3.TXT
```

If the DEL command cannot find one of the specified files, it displays a message telling you which file could not be found and continues deleting the other files.

You can also use wildcard characters to delete a group of files. For example, suppose you are closing out your accounts with a particular vendor named New Moon Supplies. Your current directory contains a variety of files that record business transactions with this vendor:

```
BUDGET.JAN
BUDGET.FEB
NEWMOON.INV
NEWMOON.ACC
NEWMOON.1
NEWMOON.2
NEWMOON.FIL
REPORT.FIL
```

To delete all files starting with the filename *NEWMOON*, type the following:

```
DEL NEWMOON.*
```

Your directory now looks like this:

```
BUDGET.JAN
BUDGET.FEB
REPORT.FIL
```

Or, to delete all of the files in your current directory, type the following:

```
DEL *
```

When you try to delete all of the files in a directory, the following message appears: "Are you sure (Y/N)?" If you type **Y** and then press **ENTER**, the files will be deleted. If you type **N**, the files will not be deleted and you will be returned to the prompt.

If you are not sure that you want to delete every file in the directory, you can use the DEL command with the /P option. With this option specified, CMD prompts you file by file, giving you the option of deleting the file or continuing with the next.

Use wildcard characters with care — remember that deletions are permanent.

Renaming a File or Directory

Occasionally, you may want to change the name of a file or directory. You can use the RENAME (REN) command to do this. For example, to change the name of the file *ADS.FIL* on drive C to *SPONSORS.FIL*, type the following:

```
REN C:\USR\DAVIDH\ADS.FIL  SPONSORS.FIL
```

You can also rename a group of files. To rename all the files in the current directory that have the extension *.TXT* so that they have the extension *.FIL*, type the following:

```
REN *.TXT *.FIL
```

When you are renaming files and directories, keep these points in mind:

- You cannot use RENAME to move a file to a different drive or directory.
- You cannot rename a file or directory by using a name that already exists.

Changing to Another Drive

To change to another drive, you type the drive letter followed by a colon (:). For example, if the current drive is drive C and you want to switch to drive D, type the following:

```
D:
```

This switches you to the current directory on drive D.

Security Attribute Considerations

Security attributes are similar to extended attributes. They are attached to individual files and directories. Consequently, when files are copied, renamed, or deleted or directories are renamed or deleted, the security attributes are affected.

The security attributes are not copied when files and directories are copied. This is true for the COPY command, XCOPY utility, and any other program that copies files. In fact, many full screen editors operate by creating a copy of a newly edited file and deleting the original file when the edit is complete. This process will delete the security attributes. Note that the MS OS/2 *MULTIUSER* System Editor will preserve the security attributes.

Security attributes are preserved when files and directories are renamed.

When files and directories are moved from one place to another on the same drive the security attributes are preserved. This operation is much like a RENAME. However, when files and directories are moved to a different drive this is like a COPY and DELETE operation; therefore, the security attributes are lost.

When files and directories are copied during the BACKUP and RESTORE process (using the MS OS/2 *MULTIUSER* BACKUP and RESTORE utilities), the security attributes are preserved.

RUNNING PROGRAMS

You can run programs such as applications and utilities from CMD. This section describes how to start a program, as well as how to set up the proper environment to run a program and how to modify standard input (input from the keyboard), standard output (output to the screen), and standard error (error messages). This section also describes what happens when you start different types of programs from CMD.

Starting a Program from CMD

To start a program from CMD, do the following:

> At the prompt, type the command that starts the program, followed by any arguments, and press **ENTER**.

For example, suppose that you want to prepare the payroll checks for your employees by using a spreadsheet program named SP. To start the program, type the program's name, in this case SP, and then press **ENTER**.

NOTE: See your application's manual for the specific command that starts the program.

Setting the PATH and Other Environment Variables

Starting a program from CMD is identical to using commands, except that you must be in the directory where the program is located or you must have set the PATH environment variable to include that directory. The PATH environment variable specifies which directories MS OS/2 *MULTIUSER* will search for programs you start. If you add your program's directory to the search path, MS OS/2 *MULTIUSER* will find and start your program regardless of the directory in which you are currently working.

NOTE: You must have the proper access to the program or directory where the program resides in order to start it.

You might also need to set other environment variables, such as DPATH, LIB, or TMP, depending on whether your program requires them. For information on setting environment variables, see "Setting Up the Environment" later in this chapter.

Starting a Program With the START Command

The START command lets you start a program in a new session. START provides an alternative to the Program Selector and also lets you start programs from a batch file. Various options to START provide you with different ways to start programs. To start a new program, type START followed by the name of the program. If you type START by itself, CMD starts running.

You can also specify a program title and options. The START command's options are described in the following sections.

Once programs start running, you can stop them at any time by pressing **CTRL+C**.

The new program starts in a background session, not in the one you are currently working in. You will need to switch to that session before you can view the program or work with it. (See "Starting a Foreground Program" later in this chapter for an alternate way to make a new program appear on your screen.)

Using START in Batch Programs

The START command is used most often in batch programs for activities such as starting programs during system start-up. If you know that you want a particular program to run every time you login, you may want to use the START command in a start-up file. For more information about batch programs, refer to Chapter 6, "Creating Simple Batch Programs." For information about start-up files, see "Using Start-Up Files" in Chapter 6.

Specifying a Session Title

To specify a session title for the new program, type the title enclosed in quotation marks immediately after the START command. This title becomes the title of the session in which the program is running.

For example, to start a program called SP with the session title "Company Spreadsheet," type the following:

```
START "Company Spreadsheet" SP
```

Using the /C Option

By using the /C option, you can cause CMD to end after a program finishes running. Normally, after a program that was started with the START command finishes running, the new session that was created does not disappear. When you specify the /C option, the new session closes after the program ends.

You can use this option to start a program without keeping an extra session with CMD running. For example, suppose you want to run the QUERY USER utility while you work on something else. Type the following:

```
START /C QUERY USER > C:QUSER.DAT
```

This starts the QUERY USER utility in another session. The QUERY USER utility obtains information about users logged into the system. This information is saved in a file named *QUSER.DAT* which is created in the current directory for drive C. When QUERY USER ends, the session in which it runs also ends. You can later view the *QUSER.DAT* file to see the information from the QUERY USER utility.

Starting a Foreground Program

By using the /F option, you can cause the new program started by the START command to be switched to the foreground. This way, when the program starts, you can see it running.

Other Options to the START Command

The START command has several other options, which are described in the following list. For details about these options, see the *Citrix MULTIUSER Command Reference.*

Option	Purpose
/K	Preserves the new session after the program ends. This is the default setting.
/N	Causes a program to start without first starting CMD.
/I	Causes the newly started session with CMD to inherit the default environment that is derived from the environment specified in the *CONFIG.SYS* file. The /I option is described in "Inheriting the Environment of the Session" later in this chapter.
/PGM	Specifies that the quoted string following this option is the name of the program to be run.

Starting a Background Program with the DETACH Command

To start programs that are not visible on your screen when they run, use the DETACH command. These background programs are processed but they do not run in a visible session. This means that you cannot provide input to them from the keyboard and they do not display output at your terminal. In addition, detached programs do not appear in the Task Selector. To detach a program, type DETACH followed by the name of the background program you want to run. For example, suppose you want to copy a set of files from one directory to another. To run the XCOPY utility as a detached program, you would type the following:

```
DETACH XCOPY PHONE ARCHIVE
```

In this example, the XCOPY utility runs as a detached program. It copies the files from the directory PHONE to the directory ARCHIVE.

Once you start a background program, MS OS/2 *MULTIUSER* displays the message "The Process Identification Number is nn" and then displays the CMD prompt again. After the prompt appears, you can continue to type commands and run programs while your background program runs.

Programs that can run without user input or that take a long time to run are good candidates for use as background programs. That way, you do not need to have CMD running in a full screen session just to start one program.

Since you cannot see the output of a background program at your terminal, you may want to redirect the output to a file or device. For example, suppose that you want an alphabetical list of a very large list of names. The following command starts a background program that reads the names from the file *NAMES*, sorts them, and writes them to the file *SORT.DIR*:

```
DETACH SORT < NAMES > SORT.DIR
```

While this is taking place, you can continue to run another program from the CMD prompt displayed on your screen. For a detailed explanation of redirection symbols (>, <, or >>) and the pipe (|), see "Redirecting Input, Output, and Error Messages" later in this chapter.

SETTING UP THE ENVIRONMENT

The environment consists of a collection of variables used by the system. These variables, called environment variables, are user-defined and can be assigned values of your choosing. Environment variables such as PATH and DPATH are commonly used by programs but you can also define your own environment variables.

When you start a program from CMD, the system searches for the program in the directories specified by the PATH environment variable. Some programs also use the DPATH environment variable, which specifies the data search path to use. If you are a programmer, you might need to set the LIB (library search path) and INCLUDE (include-file search path) environment variables.

This section describes how to set up the PATH and DPATH environment variables, as well as how to use the SET command.

Displaying Your Path

After you install your program — but before you run it — you may need to modify your PATH environment variable. The MS OS/2 *MULTIUSER* command interpreter, CMD, uses this variable to search for programs that you start from CMD. When you type a command to start a program, CMD first searches your current directory for the program. If it does not find it there, CMD searches the first directory listed by the PATH environment variable, then the second, and so on, until the program is found or all directories listed have been searched. When MS OS/2 *MULTIUSER* is first installed on your system, a default PATH setting is assigned automatically and stored in your environment along with other environment variables. You can view your path by typing the following:

```
PATH
```

NOTE: You must have the proper access to the directories in the path in order to execute programs in the path.

You can also view the values assigned to other environment variables by typing SET by itself.

You can always run a program if you are already in the directory where the program is installed or if you specify the exact path of the program on the command line. It is usually easier, however, to modify your path so that you can run your program while you are in any directory.

Setting Your Path

To modify the PATH environment variable, you can use the PATH command. To use this command, type PATH followed by the directories that you want in your search path. Separate directory names with a semicolon (;).

For example, suppose you want to add the APPS directory to your search path. Your current path is set to the OS2 directory and several of its subdirectories. You can type the following to set the new path:

```
PATH C:\OS2;C:\OS2\CTX;C:\OS2\SYSTEM;C:\APPS
```

Changes you made to your path with the PATH command are lost when you logout. MS OS/2 *MULTIUSER* reintializes your path assignment when you login.

In addition to the PATH command, you can also use the SET command to set your search path (described in the following sections).

Setting up a Data Search Path

You can set up a search path for data files by using the DPATH command. Just as the PATH command tells CMD where to locate program files, the DPATH command tells programs where to locate data files. A data file is any file not having the file extension *.EXE*, *.COM*, *.CMD*. For example, to tell MS OS/2 *MULTIUSER* to search for data files first in your current directory and then in the ACCOUNTS directory, type the following:

```
DPATH ACCOUNTS
```

NOTE: Not all programs use the DPATH variable in their data searches. If it is not used, DPATH is ignored.

Setting an Environment Variable

To set environment variables, you can use the SET command. You set environment variables by typing SET followed by the name of the variable, an equal sign (=), and the value. (For more information on environment variables, see "Setting Up the Environment" earlier in this chapter.) To view your current environment, type SET by itself. You will see a list of environment variables and the values that are assigned to them.

The form a value takes depends on what the variable is and what it is used for. For example, while the PATH variable requires directory paths, the PROMPT variable requires the special characters that define the way the prompt looks. To define or redefine an environment variable, type SET followed by the name of the variable, an equal sign (=), and the value. If you want to define an environment variable called TEMP and assign the TEMP directory on drive C to it, type the following:

```
SET TEMP=C:\TEMP
```

If you type SET by itself, the TEMP variable appears in the list.

In most cases, however, you will want to append, not override, any new values to your existing settings. For example, suppose that your current path looks like this:

```
PATH C:\OS2;C:\OS2\CTX;C:\OS2\SYSTEM;C:\OS2\INTRO
```

Also, suppose that you have just installed a spreadsheet program called SP in the directory APPS and you want to be able to run your program from any directory. You could add the APPS directory to the search path by typing the existing path, then adding the APPS directory to the end of the list. An easier way, however, is to use a variable in the SET command. To append a value to your path, type the following:

```
SET PATH=%PATH%;C:\APPS
```

The %PATH% variable is replaced by the existing path and APPS is appended to the end of the list. After you have set your path, verify that it has been added by typing SET again. For more information on setting environment variables see Chapter 6, "Using Startup Files."

Deleting an Environment Variable

To delete the current setting for an environment variable, type SET followed by the name of the variable and an equal sign (=), but do not type a value to be assigned to the variable. For example, to delete the DPATH environment variable and its current setting from your environment, type the following:

```
SET DPATH=
```

Inheriting the Environment of the Session

To start a new session with CMD and have it inherit the default environment that was set up when you started your system, use the START command with the /I option. Normally, when you type START, CMD inherits the environment of the current session. Since you can change the environment at any time by using the PATH and SET commands, your current environment may not be the same as the default environment established when the system was started. For a new session of CMD to inherit the default environment, type the following:

```
START /I
```

CMD starts to run in a new session. Now if you type SET by itself, you will see that the new environment is identical to the default environment. For more information about the START command, see "Starting a Program with the Start Command" earlier in this chapter.

Predefined User Environment Variables

Four predefined environment variables are set up by the system when you login. These variables include:

- HOME
- USERNAME
- GROUPNAME
- HOSTNAME

The home environment variable identifies your home directory. The username and groupname variables are set to the user's *username* and *groupname*, respectively. The hostname environment variable is set to the system's name.

These variables are accessed by using the percent (%) character to delimit the variable name. For example, to use the home variable, you could use the following command:

```
CD  %HOME%
```

This command changes the current directory to your home directory.

REDIRECTING INPUT, OUTPUT, AND ERROR MESSAGES

When an MS OS/2 *MULTIUSER* program runs, it usually receives input and produces output. For example, the output of the DIR command is a directory list that is usually displayed on your screen. In addition, if the system encounters errors while running a program, it displays error messages.

MS OS/2 *MULTIUSER* programs can receive input from different devices such as a keyboard or a disk file. Similarly, programs can send their output and error messages to different destinations, such as a terminal screen, a disk file, or a printer.

By default setting, MS OS/2 *MULTIUSER* programs receive input from the keyboard and send output to the screen.

Information that you type at the keyboard is called standard input (STDIN) and the information that is sent to the screen is called standard output (STDOUT). In addition, if the system encounters an error condition, it normally sends error messages to the screen. This is called the standard error (STDERR).

If you want input to come from a source other than the keyboard or have output or error messages sent to a device other than the screen, you can use two powerful features of MS OS/2 *MULTIUSER*: redirection symbols and pipes.

Redirection symbols (>, <, or >>) let you change the way that MS OS/2 *MULTIUSER* deals with input and output. Instead of taking input from the keyboard, for example, you could direct MS OS/2 *MULTIUSER* to access data from a file on your disk. Instead of displaying output on the screen, you could send the program's output or error messages to a printer.

Pipes (|) let you take the output of one program and use it as the input to another program. The uses of redirection symbols and pipes are described further in the following sections.

NOTE: Not all programs support redirection and piping. Programs must be written to support standard input (STDIN) and standard output (STDOUT).

To keep track of data and to identify files, MS OS/2 *MULTIUSER* uses file handles; some of these can be redirected. By default setting, STDIN has the value (or handle) of 0, STDOUT has the value of 1, and STDERR has the value of 2. File handles 0 through 9 can be redirected by using the MS OS/2 *MULTIUSER* command interpreter, CMD. However, file handles 3 through 9 must be assigned by a program before they can be redirected.

Pipes and redirection symbols can be used alone or in combination to change where input is read from and where output and error messages are written to. This lets you redirect input, output, and error messages to a file or device other than the keyboard or the screen.

Saving Program Output in a File

Instead of displaying messages on the screen, you can send those messages to a file on your disk. To redirect the standard output (STDOUT) to a file, use a greater-than sign (>) in your command. Unless you specify a number before the greater-than sign, the standard output (handle 1) is assumed. This means that > and 1> are equivalent.

For example, to redirect the output of the DIR command to a file named *DIRLIST.TXT*, type the following:

```
DIR > DIRLIST.TXT
```

Now, if you use the TYPE command to view *DIRLIST.TXT*, you will see that the directory list appears in the file. If the *filename* you specify does not already exist, MS OS/2 *MULTIUSER* creates a new file and stores your directory list in it. However, if the *filename* does already exist, MS OS/2 *MULTIUSER* replaces what is in the file with the new data. The old contents of the file are lost, so be sure you do not specify the name of an existing file that you want to keep.

Sending Program Output to a Device

You can also redirect standard output to a device. For example, if you want to send your directory list to a printer that is attached to your computer's first parallel port (LPT1), type the following:

```
DIR > \DEV\LPT1
```

Remember that your devices must be installed and running properly before you can redirect output to them.

Finally, if your program sends output to handles 3 through 9, you can redirect standard output to a file or device by specifying the handle number before the greater-than sign. For example, if you typed MYPROG 3> OUTFILE, this would take any output from MYPROG that is sent to handle 3 and send it to OUTFILE.

Appending Output to a File

If you want to add standard output to a file (instead of replacing the entire file), you can use two greater-than signs (>>) to tell MS OS/2 *MULTIUSER* to append the output of the command (such as a directory list) to the end of the file you specify. For example, to append your directory list to an existing file named *NEWLIST.TXT*, type the following:

```
DIR >> NEWLIST.TXT
```

When you redirect output with two greater-than signs (>>), the existing contents of the file are not overwritten.

READING INPUT FROM A FILE

If you want your program to read input from a file, use a less-than sign (<) in your command. Unless you enter a number before the less-than sign, standard input (handle 0) is assumed. This means that < and 0< are equivalent.

It is often useful to have the input for a command come from a source other than the keyboard, such as from a file. For example, suppose you have a file called *NAMES.FIL* that lists the names of several clients. If this list is not in alphabetical order, you could sort the file's contents by typing the following:

```
SORT < NAMES.FIL
```

The SORT utility takes the contents of *NAMES.FIL* as its input and displays the newly sorted contents alphabetically on the screen.

READING INPUT FROM ONE FILE AND SAVING THE OUTPUT IN ANOTHER

You can have a program both read input from a file and save output in another file by combining redirection symbols. For example, to sort the addresses in the client list in *ADDRESS.FIL* and save it in a file called *CLIENTS.ADD*, type the following:

```
SORT < ADDRESS.FIL > CLIENTS.ADD
```

Thus, *ADDRESS.FIL* is the input to the SORT utility and *CLIENTS.ADD* is the output.

SAVING ERROR MESSAGES IN A FILE

By default setting, MS OS/2 *MULTIUSER* displays error messages (STDERR) on your screen. It is sometimes helpful, however, to store them in a file. That way, you have a permanent record of any errors you encounter while running a program. Suppose you are planning to copy all of the files on drive C that contain the filename extension *.TXT* to drive D. To do this and to create a file to hold the error messages that may appear while the files are being copied, type the following:

```
COPY *.TXT D: 2> LOGFILE.MES
```

Any error messages that the operating system sends out during the copying operation are now stored in the file *LOGFILE.MES.* If you use the TYPE command to view *LOGFILE.MES,* you will see the error messages that would normally be displayed on your screen. In this case, the message might be "SYS0002: The system cannot find the file specified." You must use the number 2 before the redirection symbol to specify that the error messages, not the command's output, are redirected to *LOGFILE.MES.*

SENDING BOTH OUTPUT AND ERROR MESSAGES TO ONE FILE

If you want your log file to contain both standard output and standard error messages, you can redirect them both to the same output device or file. This can be useful when you run a detached program, since you cannot view the output from background programs while they are running. After the detached program finishes running, you can view an output file that contains output and error messages that would normally have been displayed on your screen if the program had been running in a session; for example, if you want to use the DIR command to list all of the *.TXT* files that are on your current drive. To send both the standard output (handle 1) and the standard-error messages (handle 2) to the file *LOGFILE.MES*, type the following:

```
DIR *.TXT > LOGFILE.MES 2>&1
```

In this example, "2>&1" tells MS OS/2 *MULTIUSER* that any output written to handle 2 will have the same effect as if it were written to handle 1. Therefore, the *LOGFILE.MES* file will contain both the output of the DIR command and any error messages. The order of the symbols and arguments in the command is very important here: "2>&1" must follow the name of the file.

SENDING OUTPUT AND ERROR MESSAGES TO DIFFERENT FILES

If you want output or error messages to be directed to two different files, specify the filenames separately:

```
DIR *.TXT > OUTFILE.DOC 2> ERRFILE.DOC
```

Now, any output from the DIR command is in *OUTFILE.DOC* and error messages are in *ERRFILE.DOC.*

TAKING OUTPUT FROM ONE PROGRAM AND SENDING IT TO ANOTHER

Another way you can manipulate input and output is by using pipes. Pipes work by taking the output of one command and using it as the input for the next command. Using pipes saves time, since you can run two, three, or more programs from a single command line.

Piped commands are separated by the pipe symbol (I). When you use pipes, the first command on the command line runs as usual. The output of the first command is piped (that is, used as input) to the command that follows the first pipe symbol. The second command then runs, using the piped input. This program is repeated until all of the piped commands are processed. By using pipes, you could process the same set of data in several different ways.

For example, to view a sorted directory list of the current directory, type the following:

```
DIR | SORT
```

The DIR command is carried out, and the output (the directory list) is piped to the SORT utility. Then, the SORT utility sorts the output, and the screen displays a directory list sorted by month and day.

TASKS THAT USE REDIRECTION SYMBOLS AND PIPES

You can use redirection symbols and pipes to perform many tasks. The following examples show just a few possible uses.

Sorting a Directory List and Saving it in a File

By combining the pipe and redirection symbols, you can pipe the output of one command to another command and then save the output in a file. A useful technique is to pipe two commands together and send the output to a file. For example, the following command creates a file named *DIRECT.LST* in your working directory:

```
DIR | SORT > DIRECT.LST
```

The *DIRECT.LST* file now contains a sorted list of the working directory.

You can also specify a drive other than the default drive. For example, suppose you want to create the file *DIRECT.LST* on drive C and send the sorted data to it. To do this, type the following:

```
DIR | SORT > C:DIRECT.LST
```

Printing a Sorted Phone List

As you have already seen, you can redirect both the input and the output of a program. It is sometimes useful to print the output of a program. For example, if you wanted to sort a phone list and then print out the sorted list, you could type the following:

```
SORT  < PHONE.LST >  \DEV\LPT1
```

The contents of the file *PHONE.LST* first become the input for the SORT utility and then become the output to a printer attached to LPT1.

Preventing Error Messages from Being Displayed

If you do not want error messages to be displayed on your screen, you can redirect them to NUL. Anything sent to NUL is discarded. To send error messages to NUL, type the following:

```
CMD 2> \DEV\NUL
```

This command starts a new command interpreter (CMD) and sends all error messages to NUL.

Piping Several Commands Together

You can pipe several commands together. For example, to sort several phone lists and display the sorted output one screenful at a time, you could type the following:

```
TYPE *.LST | SORT | MORE
```

All of the phone lists having the filename extension *.LST* are sent together as output to be sorted and then displayed one screenful at a time.

CHAPTER 4

USING Citrix *MULTIUSER* UTILITIES

INTRODUCTION

MS OS/2 *MULTIUSER* provides programs called utilities to help you work with directories and data files, maintain disks, print files, and display Help information.

Utilities run in a session; you can start them from the MS OS/2 *MULTIUSER* command interpreter (CMD). The examples in this chapter assume that you are in a session and have started CMD.

NOTE: All examples of *filenames* and directory names in this chapter use HPFS *filenames* and directory names.

Starting a Utility

To start a utility from the CMD command prompt, do the following:

> At the prompt, type the name of the program you want to start, followed by any command-line arguments, and press **ENTER**.

Getting HELP Information

To display Help information about warning or error messages that you may see on your screen, use the MS OS/2 *MULTIUSER* HELP batch program. Each of these messages consists of a number (for example, SYS1041) and a brief description of the error condition. For a more detailed explanation of the error condition, type HELP followed by the message number. For example, suppose that you receive the error message, "SYS0002: The system cannot find the file specified." To see a detailed explanation of this error message, type the following:

```
HELP SYS0002
```

The message number can be shortened to 0002, or just 2, so that you only need to type HELP 2 on the command line. A more detailed explanation of the possible problem and the action you might take to correct the problem appears on your screen:

```
SYS0002: The system cannot find the file
specified.

EXPLANATION: The file named in the command does
not exist in the current directory or search path
specified or the filename was entered incorrectly.
ACTION: Retry the command using the correct
filename.
```

FILE AND DIRECTORY UTILITIES

MS OS/2 *MULTIUSER* provides the following utilities to help you manage your files and directories:

Utility	Purpose
XCOPY	Copies a directory and its contents to another directory.
TREE	Displays a list of all the directories and files on a drive.
ATTRIB	Displays or sets the file flags (also known as attributes) of files.

The following sections describe these utilities.

NOTE: Because of file system security, you may not have access to all files or directories.

Copying Directories and Subdirectories

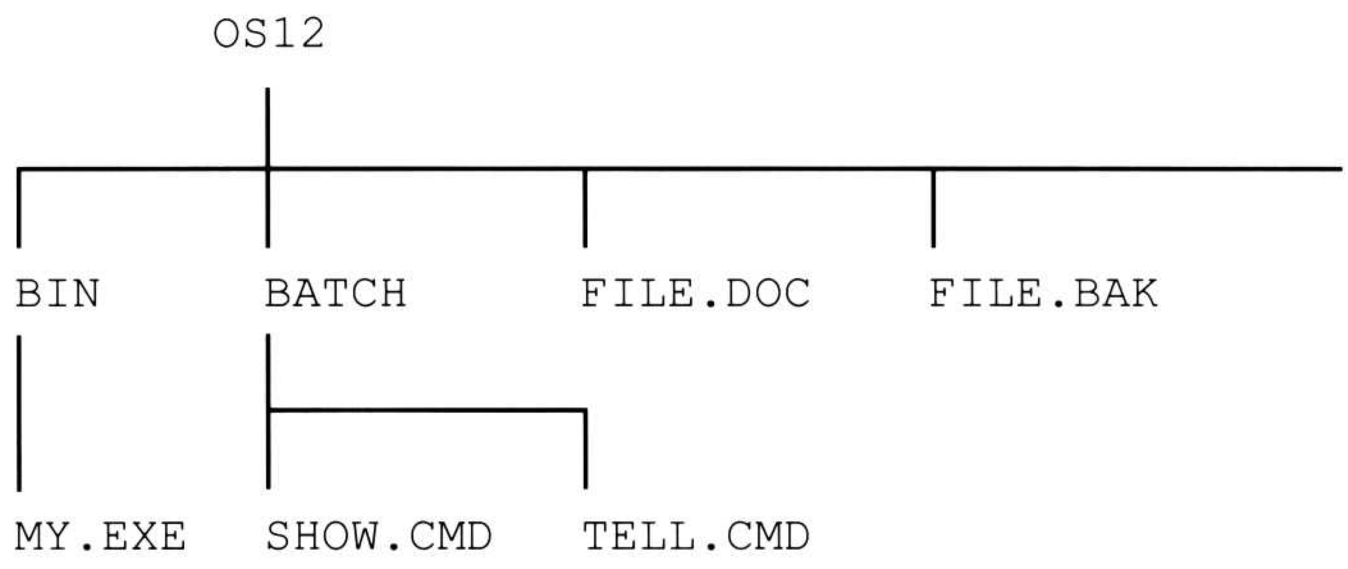
To copy an entire directory and its contents, use the XCOPY utility. To use this utility, specify a source directory that files will be copied from and a destination directory where files will be placed. The source and destination can be a drive, directory path, and/or *filename.* If you do not specify a drive or directory path, XCOPY uses the current drive and directory.

The XCOPY utility has several useful options. These options are described in the following sections.

Specifying a Directory or Subdirectory

You may want to copy not only the files in a directory but all of the subdirectories as well. You could copy the files in each subdirectory individually to the destination directory, but this is often very time consuming. An easier way is to use the /S and /E options with XCOPY. The /S option copies subdirectories unless they are empty; /E copies subdirectories that are empty. Together these options enable you to copy the files in the specified directory in addition to the files in all of the subdirectories. You must specify both options in order to copy all of the subdirectories.

To see how these options work, suppose you have a directory named OS12, which has the following subdirectories and files:



If you type the following, you will copy the files in the OS12 directory, as well as the files in the BIN and BATCH subdirectories, to the root directory in drive C:

```
XCOPY OS12\*.* /S /E C:\
```

The XCOPY utility preserves the original directory structure as it copies files and directories—the OS12 directory will appear in the root directory of drive C, BIN and BATCH will appear as subdirectories of OS12, and so on.

Confirming Each Copying Operation

The /P option causes the XCOPY utility to prompt you with "(Y/N)?" after displaying the name of each source file and before copying it. To copy the file, type **Y** for yes. To prevent that file from being copied, type **N** for no.

Verifying Each Copying Operation

You can use the /V option to verify that each file is copied accurately. This compares the destination file to the source file to make sure that they are identical.

Preserving Extended Attributes

In MS OS/2 *MULTIUSER*, some applications attach extended attributes (additional information, such as author, application type, and file history) to your files and directories. These extended attributes are used by other applications, the file system, or the operating system itself. For example, an application may check a file's extended attributes to determine whether the file is owned or created by that application.

The XCOPY utility copies a source file's extended attributes to the destination file. You can see if your files have extended attributes by using the DIR command with the /N option. If a file has extended attributes, the size of the attributes is displayed in the column before the *filename.*

You may want to use the EAUTIL utility before copying files that have extended attributes to a directory in MS OS/2 versions earlier than 1.2. By using the EAUTIL utility, you can split files and their extended attributes into separate files, copy the files, and then join the files together again. For information about the EAUTIL utility, see the *Citrix MULTIUSER Command Reference.*

If you are copying a file that has extended attributes to a file system that does not support extended attributes, you may want to use the /F option with the XCOPY utility. For more information about using the XCOPY utility on files with extended attributes, see the *Citrix MULTIUSER Command Reference.*

Security attributes are not copied.

Displaying the Directory Tree

The TREE utility displays the entire directory structure for a drive. To see the names of all the directories on your current drive (starting with the root directory), type TREE without arguments. To see the names of all the directories on a different drive, specify the name of the drive after TREE.

For example, to see the names of all the directories on drive C, type the following:

```
TREE C:
```

A list similar to this one will appear:

```
Directory path listing

Path: \ACCOUNTS

Subdirectories:    APRIL
                   MAY

Path: \ACCOUNTS\APRIL

Subdirectories: None

Path: \ACCOUNTS\MAY

Subdirectories: None
```

Since the directory-tree list may be very long, you will probably want to do one of two things when you use the TREE utility. You may want to redirect the output to a file or you may want to use the MORE utility to view the list one screenful at a time.

NOTE: Because of file system security, TREE displays only the files and directories you have read access to.

Redirecting Output to a File

Use the redirection symbol (>) to send the directory-tree list to a file. For example, to see the directories on your current drive and redirect them to the file *TREE.LST*, type the following:

```
TREE > TREE.LST
```

Then you can use the TYPE command to view the file *TREE.LST* or the PRINT utility to print the file. The PRINT utility is discussed in greater detail in "Printing Data Files" later in this chapter.

Viewing the Directory Tree One Screenful at a Time

If you have many directories, the directory-tree list may scroll on your screen too fast for you to read. To view the list one screenful at a time, you can pipe the output of the TREE utility to the MORE utility as follows:

```
TREE A: | MORE
```

Each time a screenful of information is displayed, you will see the message "-- More --" displayed at the bottom of the screen. Press the **SPACEBAR** or the **ENTER** key to display the next screenful.

Displaying Filenames

You can use the /F option after the drive letter to display the names of the files within the directories on a drive. For example, suppose you want to generate a list of all the files that you currently have on your fixed disk. To display a list of the directories and all of the files on drive C, type the following:

```
TREE C: /F
```

Displaying and Setting File Flags

You can use the ATTRIB utility to display and set file flags, which are special features that a file can have. The following file flags can be set:

- Read-only
- Archive

You have the option of turning file flags on or off. If the read-only file flag is set, you can read a file but you cannot change its contents. This flag is commonly used to prevent a file from being deleted or changed accidentally. Flagging files with the archive flag allows you to copy only recently modified files to or from a disk with utilities such as BACKUP, RESTORE, and XCOPY.

Displaying File Flags

To see which file flags are set for one or more files, type ATTRIB followed by the names of the files you are interested in. For example, to see which file flags are set on all of the files in your current directory, type the following:

```
ATTRIB *
```

A file list appears on your screen, containing letters that signify which file flags are set for the file. The flags are followed by the drive, directory path, and *filename*. The file-flag letters you might see are as follows:

- If the letter "R" appears, the read-only file flag is set for the file. This means that you can read the file, but you cannot change it.

- If the letter "A" appears, the archive file flag is set for the file. By default setting, all files have the archive file flag set when they are created or modified. Certain programs, such as XCOPY, BACKUP, and RESTORE, can turn off a file's archive file flag after they run. For more information about how the archive file flag works with these programs, see "Setting the Archive File Flag" later in this chapter.

To gain a better understanding of how file flags work, suppose that a directory contains the files *WORD.LST*, *ACCOUNTS.OLD*, and *LETTER.FIL*. When you type ATTRIB *, the following appears on your screen:

```
   R  C:\USR\DAVIDH\WORD.LST
A     C:\USR\DAVIDH\ACCOUNTS.OLD
A  R  C:\USR\DAVIDH\LETTER.FIL
```

Notice that the read-only file flags are set for *WORD.LST* and *LETTER.FIL*, so you can read these files but you cannot change them. The archive file flags are set for both *ACCOUNTS.OLD* and *LETTER.FIL* but the archive file flag is turned off for *WORD.LST*. To see the file flags for files in the current directory and in any subdirectories, type the /S option as follows:

```
ATTRIB * /S
```

When you type this command, the subdirectory file entries are listed first and the current directory entries are listed last.

Keep the following points in mind when you are copying files:

- When you copy a file by using the COPY command, the archive file flag is automatically set for the file you are copying (to the destination file). This is true even if the archive file flag is turned off for the file being copied (the source file).

- If the read-only file flag is set for the source file, the flag is not copied to the destination file.

Setting the Read-Only File Flag

The read-only file flag determines whether you can write to a file or set of files. If the read-only flag is set, it prevents a file from being accidentally modified. Use the +R option to set the read-only file flag and use the -R option to turn off the flag. Suppose, for example, that you have saved permanent personnel records in the database file *EMPLOY.DBS*. To prevent other users from modifying this file, set the read-only file flag by typing the following:

```
ATTRIB +R EMPLOY.DBS
```

When other users type DIR to get a directory list, the read-only file flag cannot be seen. However, if they try to use a text editor to modify the file, or if they try to delete the file, they will be prevented from performing the operation.

If you later want to update the personnel records, turn off the read-only file flag by typing the following:

```
ATTRIB -R EMPLOY.DBS
```

Now this file can be changed or deleted.

Setting the Archive File Flag

The archive file flag can be used as a control mechanism with the BACKUP, RESTORE, and XCOPY utilities. Use the +A option to set the archive file flag and the -A option to turn it off.

The /M option used with the BACKUP and RESTORE utilities copies only files that have their archive file flag set. The /M option then automatically turns off the archive file flag of the original files after copying them. Thus, if the archive file flag is turned off, the file is not backed up or restored.

With the XCOPY utility, you can choose to use the archive file flag in doing copying operations. If you use the /A option, only those files with their archive file flags set will be copied. If you choose the /M option, files with their archive file flags set are copied, and XCOPY automatically turns off the archive file flags of the original files after copying them. You usually will not need to turn a file's archive file flag on or off.

However, there may be times when you want to modify the way that the BACKUP, RESTORE, or XCOPY utilities operate on certain files. For example, suppose that each Monday you make a backup disk that contains specific directories by using the BACKUP utility with the /M option. To save time, you decide to copy only essential files to the backup disk. Instead of letting BACKUP find and copy all of the files that have their file flags set, you can use the -A option to turn off the file flags for all of the files that have the extension .BAK. To perform this operation, type the following:

```
ATTRIB -A *.BAK
```

This means that the files that have the extension *.BAK* will not be copied when you use BACKUP to perform the copying operation.

DATA FILE UTILITIES

MS OS/2 *MULTIUSER* provides the following utilities to help you view and work with data files:

Utility	Purpose
MORE	Displays output from a command or utility, one screenful at a time.
SORT	Sorts a file alphabetically or numerically.
FIND	Finds a text string in a file.
COMP	Compares two files or sets of files.

The MORE and SORT utilities (sometimes called filter commands) take input from a device or file, process the input, and then send it to an output device or file. As such, they are used in conjunction with redirection symbols or pipes. The MORE and SORT utilities must be run from the MS OS/2 *MULTIUSER* command interpreter (CMD). For more information on redirection symbols or pipes, see "Redirecting Input, Output, and Error Messages" in Chapter 3.

Displaying Output One Screenful at a Time

To display output one screenful at a time, use the MORE utility. You can use this utility in two ways:

- To view the output of a command or utility one screenful at a time.
- To view files one screenful at a time.

You can either redirect an input file or pipe the output of a command or utility to MORE. When the first screen is filled, the message "-- More --" appears at the bottom of the screen. Press the **ENTER** key to display the next screenful of information. To view the output of a command or utility one screenful at a time, redirect the output to the MORE utility. To do this, use the pipe (|) symbol.

For example, a long directory list often contains more *filenames* than can appear in a single screen. You can use the MORE utility to display your directory list one screenful at a time by typing the following:

```
DIR | MORE
```

Other commands and utilities, such as TYPE, display output information that can also be piped to the MORE utility. For example, suppose that you have a long file, *CLIENTS.NEW*, that you want to view on your screen. The following command pipes the file to MORE:

```
TYPE CLIENTS.NEW | MORE
```

When you type this command, the *CLIENTS.NEW* file is displayed one screenful at a time. You could accomplish the same thing by redirecting a file as input to the screen, one screenful at a time. To do this, type the following:

```
MORE < CLIENTS.NEW
```

Sorting Input and Output

To sort input or output according to the character found in a specified column, use the SORT utility. Most often you would want to sort by the character in the first column — for example, by the first character of the first word on each line—but you can sort by any column you want. The character is sorted based upon its location in the character set you are using. Which character set you are using depends upon the code page that is set up for your system.

For the United States character set, the characters are sorted from 0 to 9 and then from A to Z. The SORT utility does not distinguish between uppercase and lowercase letters. You can sort files of up to 64K in length.

You can use the SORT utility in two ways:

- To sort a file that has been sent as input to SORT.
- To sort the output of another command or utility.

Sorting a File

To sort the lines in a file, you redirect an input file to sort. Each line in the file is sorted based upon the character found in the first column, unless you specify a column number. (For more information on how to specify a column number to be used for sorting, see "Sorting by Column" later in this chapter.) For example, suppose that the file *INFILE.LST* contains the following list of customer names:

Draper, John
Benson, Clara
Wiggins, Bess
Peters, Marcus

To sort the contents of the *INFILE.LST* file, type the following:

```
SORT < INFILE.LST
```

The list is then sorted and displayed in this way:

```
Benson, Clara
Draper, John
Peters, Marcus
Wiggins, Bess
```

NOTE: The SORT utility does not permanently change the order of the contents in the original file. For more information about saving the sorted output in another file, see "Redirecting Input, Output, and Error Messages" in Chapter 3.

If you want to reverse the order so that the file is sorted from Z to A or from 9 to 0, you can use the /R option. For example, if you use this option with *INFILE.LST*, your sorted file will look like this:

```
Wiggins, Bess
Peters, Marcus
Draper, John
Benson, Clara
```

Sorting Output

You can also pipe the output from another command or utility to the SORT utility. For example, to sort the contents of *INFILE.LST* with the TYPE command, type the following:

```
TYPE INFILE.LST | SORT
```

You will see the *INFILE.LST* file sorted and displayed in alphabetical order.

```
Benson, Clara
Draper, John
Peters, Marcus
Wiggins, Bess
```

Sorting by Column

You can specify any column to be used for the sorting. The option /+*N* sorts according to the character in column *N*. For example, to sort your directory list according to the day of the month it was created or last modified, you could sort according to the day of the month field. Let's say that field starts in column 4 of the directory list. To sort by the date field, type the following:

```
DIR | SORT /+4
```

Your directory list will now appear like this:

```
   6 File(s)    841728 bytes free

5-10-89   11:16a      56  0   WORD.LST
4-16-89    8:26a     120  0   CAT.TXT
3-20-89    4:10p     252  0   LETTER.FIL
3-25-89    3:37p    <DIR> 0   .
3-25-89    3:37p    <DIR> 0   ..
6-25-89    5:52p      92  0   ACCOUNT.OLD

 The volume label in drive C is APPS.
 The Volume Serial Number is A512:6414

 Directory of C:\USR\DAVIDH\CLUB
```

Searching for a String in a File

To search for a string in a file, use the FIND utility. If FIND locates the string it is searching for, it displays the name of the file followed by the line that contains the string. To use the FIND utility, type FIND followed by any options, the string you are searching for (in quotation marks), and the name of the file to search. You cannot use wildcard characters (* or ?) in a filename specification.

Because the FIND utility is case-sensitive, you must type the string with the exact uppercase and lowercase characters that you want to search for. If you do not specify a filename for the search, standard input is used—this means that, unless input has been redirected from a file or from the output of another program, the FIND utility reads input from the keyboard. (You terminate keyboard input by pressing **CTRL+Z** or **F6** and then pressing **ENTER**.)

For example, to search for the last name "Smith" in the *PHONE.LST* file, type the following:

```
FIND "Smith" PHONE.LST
```

If you are searching for a string that contains quotation marks (for example, "Hi there!" or I said "Hi there!" to Ralph), you must substitute two sets of quotation marks for the single set of quotation marks, as in the following examples:

```
FIND """Hi there!""" TEST.DOC
```

or

```
FIND "I said ""Hi there!"" to Ralph" TEST.DOC
```

Although you cannot use wildcard characters to specify filenames to be searched, you can pipe another command to the FIND utility. For example, if you want to search the files *PHONE.MKT*, *PHONE.ENG*, and *PHONE.SUP* for the name "Smith," pipe the output of the TYPE command to the FIND utility as follows:

```
TYPE PHONE.* | FIND "Smith"
```

This way, the FIND utility searches through three files instead of one. You can also specify more than one file to search by listing each file separately after the search string.

Displaying Lines That Do Not Contain a String

If you want to search a file and display those lines that do not contain the specified string, use the /V option. For example, suppose that you are searching for client names that are listed in the file *CLIENTS.LST.* To locate the client names that do not have the string "Past Due" after their names, type the following:

```
FIND /V "Past Due" CLIENTS.LST
```

Displaying the Count of Lines That Contain a String

To display the count of lines that contain the specified string, use the /C option. For example, you could find the count of lines that contain the string "Past Due" by typing the following:

```
FIND /C "Past Due" CLIENTS.LST
```

If you specify /C with /V, the FIND utility displays the count of lines that do not contain the string you typed.

Displaying the Line Number

To display the line number of any lines that contain a specified string, use the /N option. The /N option causes each line to be preceded by its line number in the file.

If you specify /C with /N, the FIND utility ignores /N.

Comparing Two Files

To compare one file or set of files with a second file or set of files, use the COMP utility. The files can be on different drives or in different directories. To use the COMP utility, type COMP followed by the drive, directory path, and *filename* of both files. If you do not specify a directory path or *filename*, COMP assumes that the files are in the current directory on the current drive.

You can use the COMP utility in several ways.

To compare the contents of drive A with drive C, type the following:

```
COMP A: C:
```

To compare the contents of the PROFITS directory on drive A and the PROF directory on DAVIDH's home directory on drive C, type the following:

```
COMP  A:\PROFITS  C:\USR\DAVIDH\PROF
```

To compare the contents of two files, \USR\DAVIDH\1988\SAMPLE.C and \USR\DAVIDH\1989\SAMPLE.C, in different directories, type the following (on the same line):

```
COMP C:\USR\DAVIDH\1988\SAMPLE.C
C:\USR\DAVIDH\1989\SAMPLE.C
```

To compare the files in DAVIDH'S home directory on drive C that have the extension *.ASM* with the files in the root directory on drive B that have the extension *.BAK*, type the following:

```
COMP  C:\USR\DAVIDH\*.ASM  B:\*.BAK
```

If the files being compared are of different sizes, COMP displays a message telling you of their size differences and then asks you if you want to continue. If you type **Y** for yes, COMP displays the location and contents of any mismatched bytes. After 10 mismatches occur, the comparison stops and you are asked whether you want to compare two more files. An error message appears if COMP does not find an end-of-file (**CTRL+Z**) marker in a file.

PRINTING DATA FILES

The PRINT Utility

Use the PRINT utility to print any MS OS/2 *MULTIUSER* data file. To use the PRINT utility, type PRINT followed by a *filename*. By default setting, PRINT sends the file to the printer that is attached to your computer's LPT1 port. If you want to send the file to a printer that is attached to another port, use the option /D:*device*. For example, to print the file *REPORT.FIL* on the parallel printer that is attached to LPT2, type the following:

```
PRINT /D:LPT2 REPORT.FIL
```

Remember that LPT2 and the printer that is attached to it must be set up correctly before you can print to it.

The PRINT utility can be used by itself or in conjunction with the print spooler.

Spooler Considerations

If the spooler is enabled for your system, you can use the SPOOL utility to control and query the status of any of your print jobs you have waiting in a print queue.

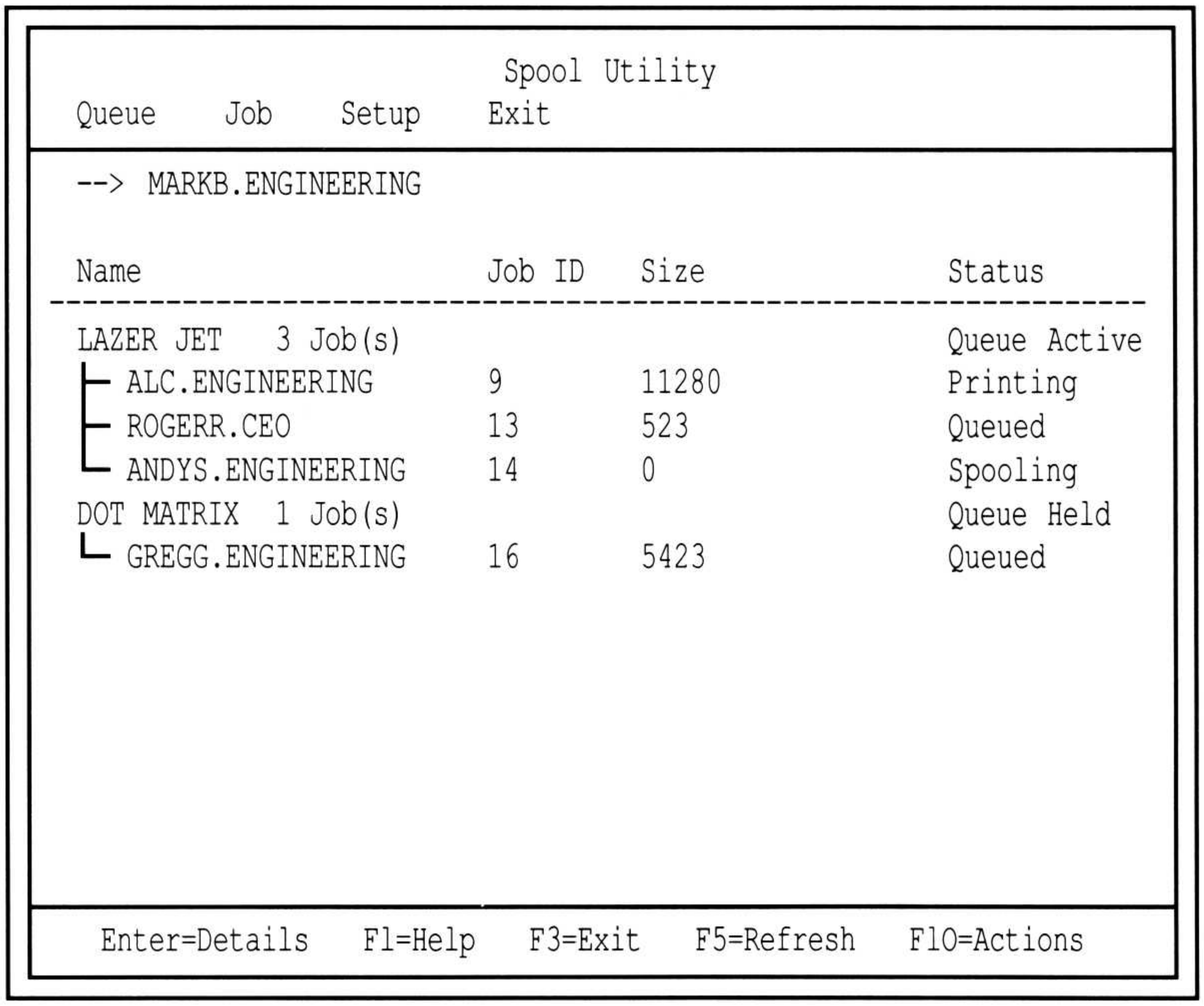
With the SPOOL utility you can do the following:

- Query all print jobs in the system. Note that the Guest class may only query its own print jobs.
- Hold your print job.
- Release your print job.
- Cancel your print job.
- Start your job again.
- Change your printer mode.
- Cancel all your jobs in a print queue.

To do any of the following, start the SPOOL utility. From the command line type:

```
SPOOL
```

A list of print jobs in all print queues will be displayed. If you are a Guest, only your print jobs will be displayed. An example of the listing follows:



```
                              Spool Utility
Queue     Job     Setup     Exit

--> MARKB.ENGINEERING

Name                        Job ID    Size                Status
-------------------------------------------------------------------
LAZER JET    3 Job(s)                                     Queue Active
├─ ALC.ENGINEERING          9         11280               Printing
├─ ROGERR.CEO               13        523                 Queued
└─ ANDYS.ENGINEERING        14        0                   Spooling
DOT MATRIX   1 Job(s)                                     Queue Held
└─ GREGG.ENGINEERING        16        5423                Queued

  Enter=Details    F1=Help    F3=Exit    F5=Refresh    F10=Actions
```

Each of these operations can be performed on the command line, bypassing the menu panels. For more information on the SPOOL utility, see the *Citrix MULTIUSER Command Reference.*

The following SPOOL utility options are only available if the selection bar is on a print job. They are unavailable if the selection bar is on a queue. The following options will only work if the highlighted job belongs to you (your *loginname*).

To Hold Your Job

Use the following steps to pause your print job so that it will not be printed until it is released.

1. Move the selection bar to the job you wish to hold.
2. Press **F10** to move the cursor to the action bar.
3. Select Job and press **ENTER**.
4. Select Hold Job and press **ENTER**.

NOTE: This is not selectable from the menu if the job is already held.

To Release Your Job

Use the following steps to release your print job that is being held so that it is available for printing.

1. Move the selection bar to the job you wish to release.
2. Press **F10** to move the cursor to the action bar.
3. Select Job and press **ENTER**.
4. Select Release Job and press **ENTER**.

NOTE: This is not selectable from the menu if the job is not held.

To Cancel Your Job

Use the following steps to remove your job from the print queue.

1. Move the selection bar to the job you wish to cancel.
2. Press **F10** to move the cursor to the action bar.

3. Select Job and press **ENTER**.

4. Select Cancel Job and press **ENTER**.

To Start Your Job Again

Use the following steps to stop your print job from printing and start printing it over from the beginning.

1. Move the selection bar to the job you wish to start again.

2. Press **F10** to move the cursor to the action bar.

3. Select Job and press **ENTER**.

4. Select Start Job Again and press **ENTER**.

NOTE: This is not selectable from the menu if the job is not printing.

To Change Your Printer Mode

You may change two settings of the printer for your print jobs:

- Characters per line.

- Lines per inch.

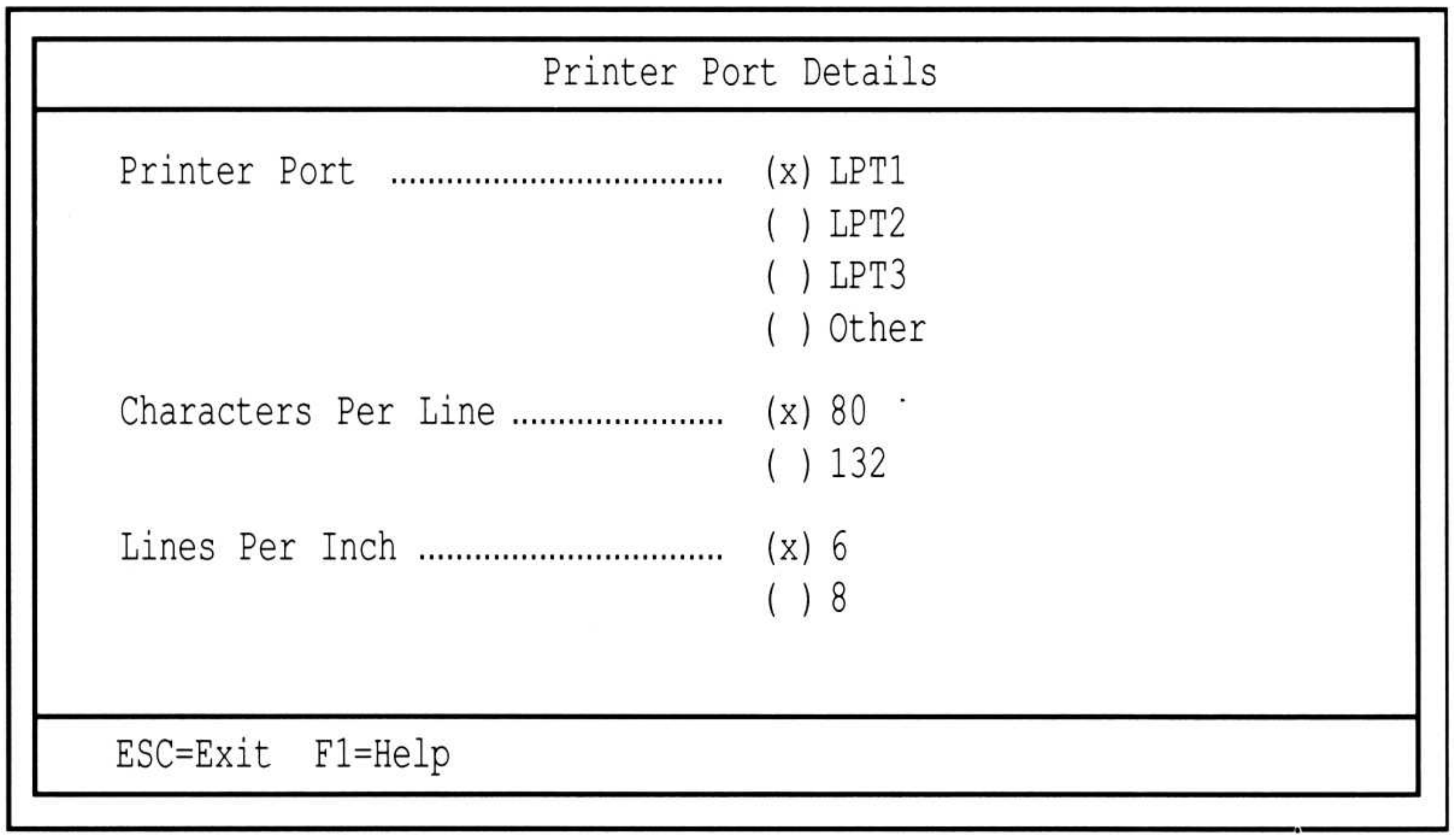
Use the following steps to change your printer mode:

1. Press **F10** to move the cursor to the action bar.

2. Select Setup and press **ENTER.**

3. Select Printer Ports and press **ENTER**.

4. The following window is displayed:



```
                         Printer Port Details

  Printer Port ..................................  (x) LPT1
                                                   ( ) LPT2
                                                   ( ) LPT3
                                                   ( ) Other

  Characters Per Line ......................       (x) 80
                                                   ( ) 132

  Lines Per Inch ...............................   (x) 6
                                                   ( ) 8

  ESC=Exit  F1=Help
```

5. Use the cursor to select the printer port details you want and press **ENTER** to set them.

The following SPOOL utility option is only available if the selection bar is on a print queue. It is unavailable if the selection bar is on a print job.

To Cancel All Your Jobs in a Print Queue

Use the following steps to cancel all of your print jobs from a print queue.

1. Move the selection bar to the queue that contains the jobs you wish to cancel.

2. Press **F10** to move the cursor to the action bar.

3. Select Queue and press **ENTER**.

4. Select Cancel All Jobs in Queue and press **ENTER**.

TURNING ANSI SUPPORT ON OR OFF

The ANSI utility turns the ANSI extended screen and keyboard support on or off. If ANSI support is turned on, the program you are using (or creating) can use the ANSI escape sequences. An ANSI escape sequence is a series of characters (beginning with an escape character or keystroke) that you can use to define MS OS/2 *MULTIUSER* functions. When you login, the default setting for ANSI support is on. To turn ANSI support off, type the following:

```
ANSI OFF
```

To turn ANSI support back on, type the following:

```
ANSI ON
```

See Appendix B in the *Citrix MULTIUSER Command Reference* for a description of the ANSI Escape Sequences.

CHAPTER 5

USING THE MULTIUSER FEATURES OF Citrix *MULTIUSER*

INTRODUCTION

This chapter describes some MS OS/2 *MULTIUSER* utilities and features that you may need to use in the MS OS/2 *MULTIUSER* multiuser environment. As a member of the User security class, you have permission to run these utilities as described.

SENDING MESSAGES TO OTHER USERS

The message utility, MSG, lets you send notes to other users who are currently logged in. To determine if a user is currently logged in, use the QUERY USER utility to display a list of *loginnames* that are currently logged in. When you use MSG, you identify the users that you wish to receive your message.

You use MSG at the CMD command prompt. In the following example, the message "Let's meet as soon as possible to discuss those work items." is sent to the login belonging to the *loginname* ZEUS.WRKGRP. Enter the following at the command prompt (on one line):

```
MSG ZEUS.WRKGRP Let's meet as soon as possible to
discuss those work items.
```

Both the message and the *loginname* of the sender is displayed in the login of the receiver.

For more information on the MSG utility and its options, refer to the *Citrix MULTIUSER Command Reference.*

UNDERSTANDING SECURITY

This section provides details of security concepts, including security classes, permission, and Access Control Lists (ACLs). It also provides several task oriented descriptions of utilities a user can access to control his or her security environment. Prior to reading this section you should review the security introduction provided in Chapter 2.

User Class and Guest Class

Classes are used to control who has access to system functions and utilities. In a standard installation, you would most often be placed in the User class. This generally allows you complete query and update control over your own environment but no ability to query or change information about other users.

You may be placed in the Guest class. A Guest has query capabilities similar to User class; however, a Guest cannot change any configuration data.

Guests and Users may be further restricted by the System Administrator. Such restrictions as when you can login, what terminals are accessible, and what applications can be run are controlled by the System Administrator.

Security Permissions

The following describes all of the security permissions associated with the different types of resources that can be secured in MS OS/2 *MULTIUSER*. Each permission is given a unique ID in the form of a single letter. This letter is used as a shorthand in the utilities that query and set permissions. The following are two additional shorthands applying to all resource types:

ID	Name	Description
*	All	This ID gives you all of the permissions that apply to the particular type of resource being targeted. This is only a shorthand method used by the utilities and is not maintained internally.
N	Null	This indicates that you have NO permissions. This equates to the absence of all permissions, and is a method of excluding access.

File System Permissions

File system permissions apply to files and directories. These permissions have slightly different meanings depending on what the target resource is, a file or a directory. If the target resource is a file with no ACL, the file's permission is governed by its implicit ACL that comes from a parent directory entry.

ID	Name	Description
C	Create	Provides the ability to create a file or subdirectory. Once a file is created, it can be written to until the file is closed, at which time Write (W) permission is required for further updates.
D	Delete	Provides the ability to delete a file or remove a subdirectory. Delete permission on a directory entry does not grant permission to delete that entry.

ID	Name	Description
W	Write	Provides the ability to update a file. This does not grant a program you are running permission to truncate a file, since file truncation is considered to be a create operation.
R	Read	Provides the ability to read and copy a file, change to a directory, or view the contents of a directory. This does not grant Execute (X) permission to a file.
X	Execute	Provides the ability to execute the program. This does not allow you to read or copy the program.
A	Attributes	Provides the ability to change the file or directory attributes.

Device Permissions

ID	Name	Description
U	Use	Provides the ability to use the device.
V	Reserve	Provides the ability to reserve the device for exclusive use.

Terminal Permissions

ID	Name	Description
L	Login	Provides the ability to login using the target terminal.

Security Attributes

Named resources in MS OS/2 *MULTIUSER* can have certain security attributes. These attributes are similar to file system extended attributes and are used to describe various security characteristics of the resource.

Only HPFS partitions support security attributes. Because MS OS/2 *MULTIUSER* is always installed on an HPFS partition, security attributes can be maintained for almost all named resources. However, these security attributes cannot be maintained for files and directories that reside on a file system other than HPFS.

There are three types of security attributes: Access Control Lists, Audit Attributes, and Ownership. The owner is the *loginname* which created the file or the name specified when changed by the OWNER utility. Only the owner or an Administrator class user can modify the permissions.

Access Control List (ACL)

One security attribute is called an Access Control List (ACL). The purpose of an ACL is to maintain a list of users and their associated access permissions for the resource to which the ACL attribute is attached. By providing a list, you can selectively control access to your files and directories.

An entry in an ACL contains two pieces of information: the identification of the login and the permissions associated with the login. The permissions indicate what types of operations (like read or write) are permitted and the identification relates to a specific *loginname* or a group of logins.

Several ACL concepts simplify the definition and maintenance of security. These are:

- Implicit file system ACLs. An MS OS/2 *MULTIUSER* file system is a hierarchy of directories, subdirectories, and files within subdirectories. MS OS/2 *MULTIUSER* provides a file system security where all files and directories that do not have ACLs are governed by the parent access controls. These files and directories are said to have an implicit ACL.

- Global access names. Within an ACL there is a list of *loginnames.* This allows for granting access to a file or directory to a select set of system users. To simplify the maintenance, the *loginname* can contain a global specification for the *username* and the *groupname.* For example, an entire group of users can be given equal access permissions to a resource by entering "*.groupname" as the *loginname.*

- Full screen ACL maintenance. The CONFIG ACCESS utility is the mechanism for controlling the file system resource access. This utility provides a full screen interface to visually describe who has what type of access and allows you to make the necessary changes without intimate knowledge of the ACL concept.

What Happens When You Are Denied Access

You can be denied access to resources for a number of reasons, all of which are related to the resource you are trying to access and the permissions you have. Correcting the problem usually requires the resource owner or System Administrator to change your permissions. This is done using the CONFIG ACCESS utility. In addition, you may be asked, as owner, to give another user permission to access your data. These operations are described later in this chapter.

Access Denied Error

The message you get when you are denied access due to security permissions is "SYS0005 Access Denied." However, depending on the application, you may get other file related errors. If you are unsure if an application is failing due to security, ask your System Administrator to audit you. This audit will provide security related data.

Cannot Find File

Paths and files that you do not have access to will be skipped over when the system is processing a "Find" operation. This will occur when handling PATH and DPATH operations. The result is that your application cannot find a file or directory. This will not show up as access denied and will not be in an audit log.

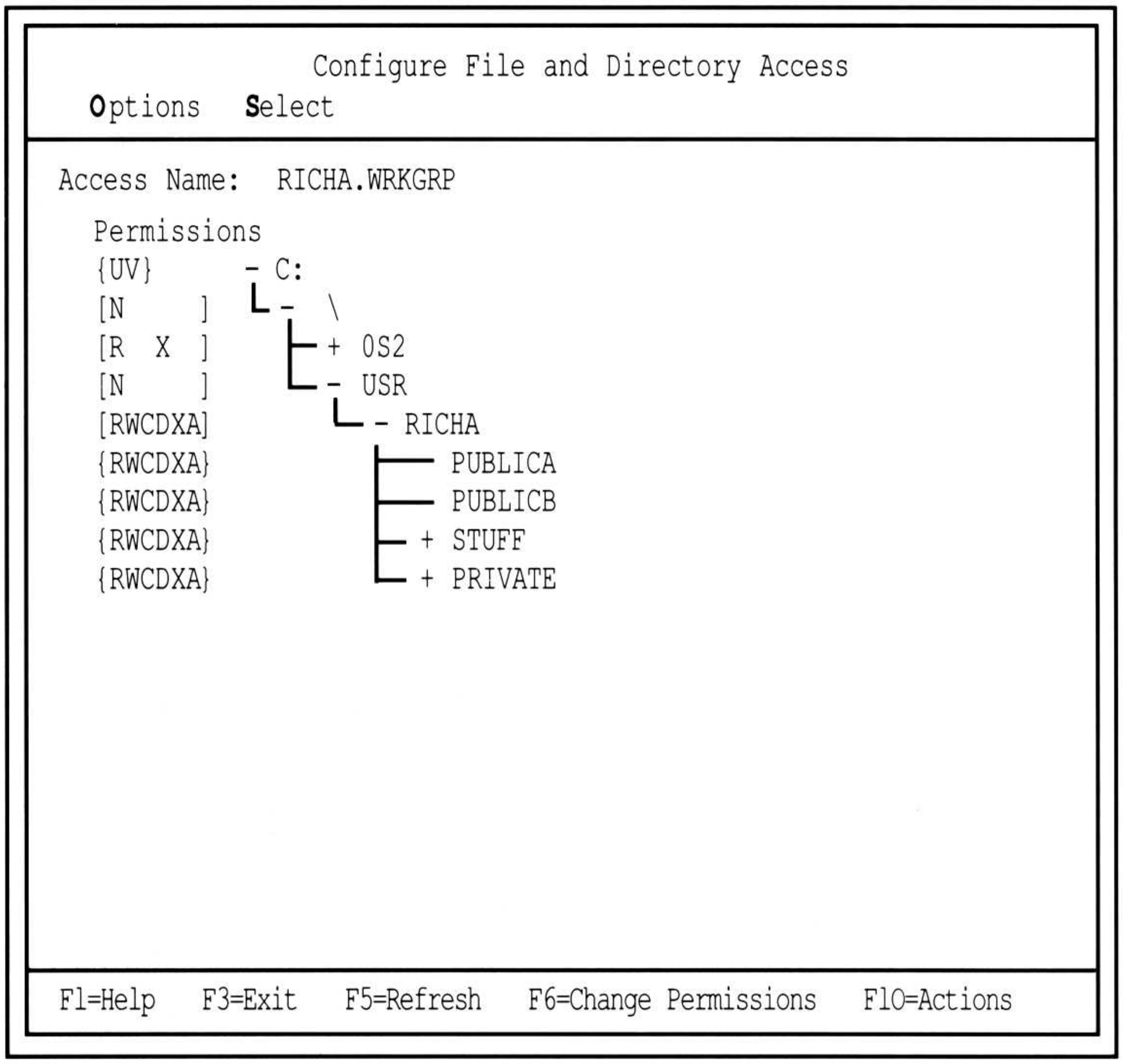
Popup When Executing a Program

On rare occasions, you may get a popup indicating that a dynalink library (DLL) is in error. The popup will list the error as SYS0005. This means your program or dynalink is not authorized to complete the load process. Your System Administrator may need to register the program.

Displaying the Files And Directories You Have Access To

Use the following procedure to display the directories you have access to.

Type in CONFIG ACCESS and press **ENTER**. A screen similar to the following is displayed.



```
                Configure File and Directory Access
  Options   Select

Access Name:  RICHA.WRKGRP
  Permissions
  {UV}      - C:
  [N     ]  └- \
  [R  X  ]    ├─+ OS2
  [N     ]    └─- USR
  [RWCDXA]        └─- RICHA
  {RWCDXA}           ├──── PUBLICA
  {RWCDXA}           ├──── PUBLICB
  {RWCDXA}           ├── + STUFF
  {RWCDXA}           └── + PRIVATE

F1=Help   F3=Exit   F5=Refresh   F6=Change Permissions   F10=Actions
```

From this screen you can:

- "Expand" directories which have a "+" or "Compress" directories which have a "-". You can view the permissions from explicit ACLs "[...]" or implicit ACLs "{...}".
- View the files and its permissions within a directory.
- Change the Access Name and see what permissions that access name has on the displayed directory structures.
- Refresh the screen.
- View or change an ACL for a specified directory.

- Change the permissions of several directories for one Access Name by marking the desired directories.

Press the **F1** key for help and instruction on any of these operations.

Giving Other Users Access to Your Files and Directories

Use the following procedure to give ANDYS Read (R) access to your directories PUBLICA and PUBLICB.

1. Type CONFIG ACCESS and press **ENTER**.

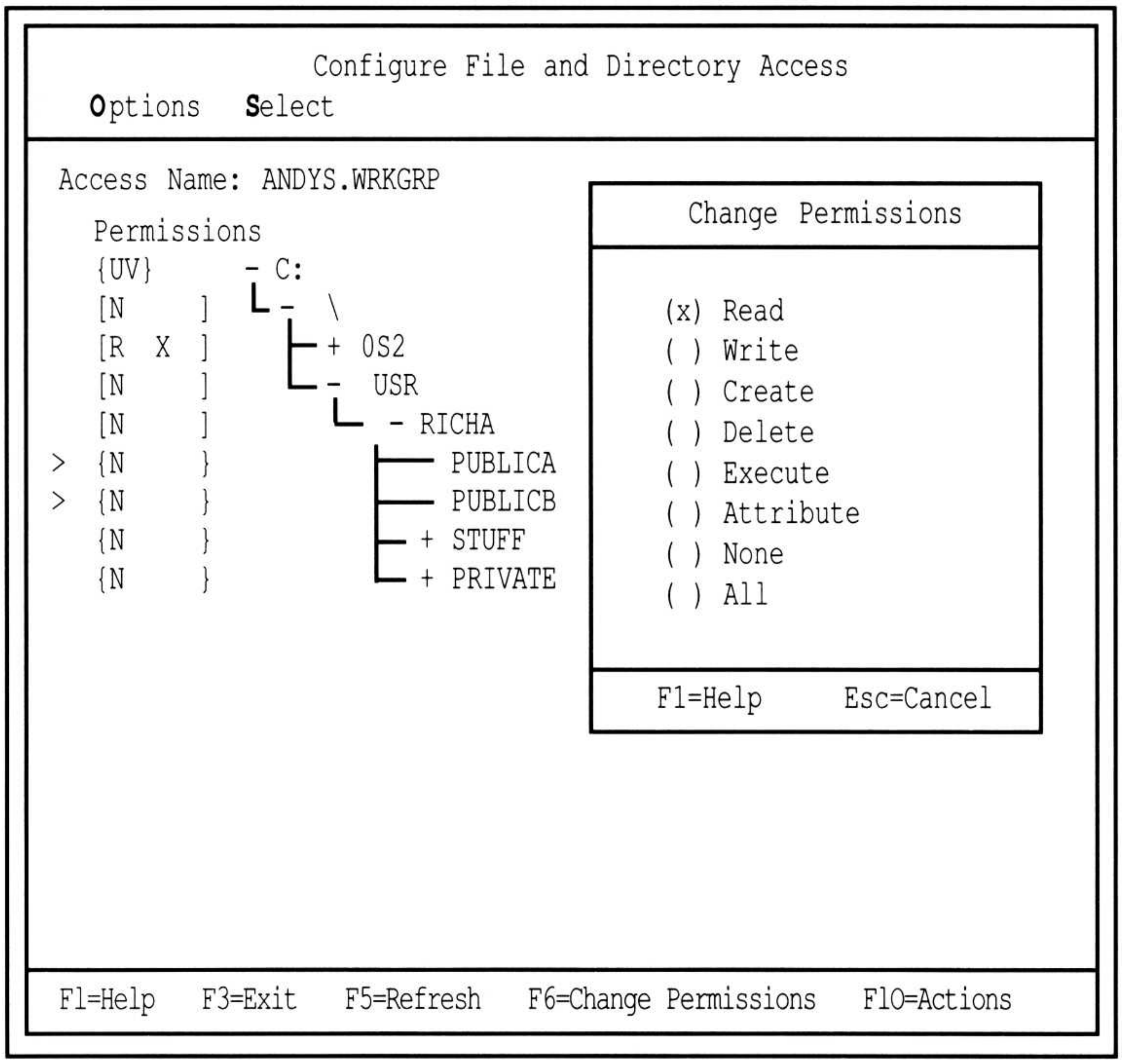
2. Use the Options pulldown "Change Access Name" to change the Access Name to "ANDY.WRKGRP".

3. Expand the directory to view the RICHA's subdirectories. (You may have to use "Expand All" in cases where you don't have access permission to all subdirectories in a path.)

4. Cursor to directories PUBLICA and PUBLICB and press **SPACEBAR**. A ">" will indicate that these directories have been selected.

5. Press **F6** to display the "Change Permissions" window. Select "(X) Read" and press **ENTER** to execute.

The following screen illustrates points described in these steps.



```
                    Configure File and Directory Access
   Options   Select

 Access Name: ANDYS.WRKGRP
                                                  Change Permissions
   Permissions
   {UV}       - C:
   [N    ]  └ -  \                              (x) Read
   [R  X ]    ├ + OS2                           ( ) Write
   [N    ]    └ -  USR                          ( ) Create
   [N    ]       └  - RICHA                     ( ) Delete
 > {N    }           ├── PUBLICA                ( ) Execute
 > {N    }           ├── PUBLICB                ( ) Attribute
   {N    }           ├ + STUFF                  ( ) None
   {N    }           └ + PRIVATE                ( ) All

                                                F1=Help      Esc=Cancel

 F1=Help   F3=Exit   F5=Refresh   F6=Change Permissions   F10=Actions
```

Gaining Access to a Serialized Device

Some devices, such as drive A and drive B, are set up by the Administrator to be serialized. To do this the Administrator will give you reserve (V) access. In this case, you will be denied access until you reserve the device.

For example, you have V access to drive A and need to copy a file to a diskette.

1. Type RESERVE A: and press **ENTER**.

   If the device is not already reserved, the utility will reserve drive A for use by your *loginname*.

2. Copy your file to drive A.

3. Type RESERVE A: /D and press **ENTER**.

   This frees drive A so that others can use it.

Giving Ownership of Your File or Directory to Someone Else

Ownership of a directory or file allows you to control the permissions. There may be instances where you need to transfer ownership to someone else.

For example, you manage a shared directory called \REPORTS. You change jobs and want to allow another person (ANDYS) to manage the directory.

To do this, type:

```
OWNER \REPORTS /C:ANDYS  /S
```

and press **ENTER**.

This gives ANDYS ownership of all directories and files at and below \REPORTS, including the directory \REPORTS.

USING DISCONNECTED LOGINS

Disconnecting Your Login

When you login to MS OS/2 *MULTIUSER*, a login is created for your *loginname*. The login is connected to the terminal where you logged in. Programs running in your login interact with you through the screen and keyboard of the terminal.

A disconnected login is a login that is not connected to a terminal. Programs in a disconnected login run in the background. MS OS/2 *MULTIUSER* automatically disconnects a login from its terminal when it detects that the terminal has lost power or has timed out, and if the *loginname* has been configured for the auto-disconnect feature. You can use the DISCONN utility to disconnect your login from the terminal if you need to leave your terminal but want to keep your programs running.

Disconnect (DISCONN) Example

Use the DISCONN utility at a CMD command line. Type DISCONN and press **ENTER**. In the following example, DISCONN disconnects the current login.

```
DISCONN
```

After you disconnect your login, the Login: prompt appears at your terminal and your login is running in the background.

Connecting Your Disconnected Login

When you login to the system through the Login: prompt, the system automatically connects your disconnected login to the terminal where you are logging in, even if the terminal is different from the one where the disconnected login was originally located.

The System Administrator may configure your user profile so that your login must reconnect to the same terminal where it was originally located. If this is the case, you must perform the login procedures at that terminal.

If you cannot login at the same terminal, you can use the CONNECT utility to connect your disconnected login to the new terminal. First perform the login procedures at your terminal, then use the CONNECT utility. The CONNECT utility connects your disconnected login to your terminal.

CONNECT Example

After logging in, use the CONNECT utility at the CMD command line. Type CONNECT and press **ENTER**. Your disconnected login is connected to your terminal.

```
CONNECT
```

You use the **CTRL+TAB** hotkey to switch between the login that you connected and the login that was created when you logged in.

CHANGING YOUR PASSWORD

Login password protection is a security mechanism provided to ensure the security of files and data. Depending on the level of system security in your operating environment, it may become necessary to change your *password* periodically. It is generally a good idea to change your *password* often to avoid unauthorized access to your files and data. You may change your *password(s)* in one of two ways.

First, you may change your *password* at the Login: prompt by supplying a new *password* and verifying it at login time. An example follows:

```
Login: your login name
Password: oldpassword/newpassword/newpassword
```

The forward slashes (/) are required to delimit the old *password* and new one. You must type the new *password* twice: once to specify it and once to verify that it is typed correctly. When you type your *password*, it will not be echoed to the screen. Instead, underscores will be echoed, a security feature used to inhibit password theft. When changing your *password*, avoid trivial *passwords* such as your name or date of birth. *Passwords* may consist of any alphanumeric combinations, unless your system environment dictates otherwise.

NOTE: You cannot use the forward slash (/) in your *password*, because it is used to delimit *passwords* when changing them.The password is case sensitive; hence, password ABCDEF is not the same as abcdef or ABCdef.

If your *password* expires (in system configurations that require periodic changes) you will have to change your *password* using the method described above.

A second method to change your password is by using the PASSWORD command. You can invoke this command at any time from the command prompt. For example, type the following:

PASSWORD /PASSWORD:*old*/*new*/*new*

In the example above, *old* is your old password and *new* is the new password that you want.

Another way to change your password is to type PASSWORD by itself and press **ENTER**. The screen prompts you for the change password information as follows:

Screen Displays	You Type and	Press
`[C:\USR\DAVIDH]`	PASSWORD	**ENTER**
`Old password:`	*oldpassword*	**ENTER**
`New password:`	*newpassword*	**ENTER**
`New password:`	*newpassword*	**ENTER**

Your *password* is immediately changed and you must use the new *password* on your next login attempt.

If you receive the "Invalid password" message, it could be due to one of many errors. Typing errors are usually the cause of this error; you may have typed your *loginname* incorrectly, typed your old *password* incorrectly, or typed your new *password* and its verification incorrectly. Carefully re-enter your *loginname* and *password.*

Your system may be configured to accept only non-trivial *passwords.* In this case, contact your System Administrator to obtain the correct *password* syntax.

CHAPTER 6

BATCH FILES

INTRODUCTION

Once you know how to use MS OS/2 *MULTIUSER* commands, you can create your own programs to carry out complex or specialized tasks. With MS OS/2 *MULTIUSER*, you can combine commands and programs into a single program called a batch program or batch file.

With batch programs, you can do the following:

- Run several commands from one batch program.
- Carry out the same batch program with different data.

You can use batch programs to simplify a series of repetitive tasks, thus reducing the amount of typing you need to do. Or you can create custom commands designed for your particular work needs. You might, for example, need to copy different directories on a regular basis from your fixed disk to a backup floppy disk. Instead of copying each directory separately by typing five COPY commands at the prompt, type the name of the batch program once and let it do the five copying operations for you.

A batch program consists of one or more lines of instructions that the command interpreter (CMD) reads and processes one at a time. As each line is processed, it is displayed (echoed) on your screen. (Later in this chapter you will learn how to prevent each line from being echoed.)

The first sections of this chapter demonstrate how to build a simple batch program to perform a series of tasks. In later sections, you will learn how to improve and enhance this program with advanced batch-programming techniques.

NOTE: All examples of file and directory names in this chapter are for the High-Performance File System (HPFS) naming features.

RUNNING A BATCH PROGRAM

To run your program, type the name of your batch file, without the extension, then press **ENTER**. For example, if your batch file is named *SPECIAL.CMD*, type SPECIAL at the command prompt to run the batch program. If you want to stop your program before it finishes running, press **CTRL+C**.

USING BATCH COMMANDS

Batch programs have a *.CMD* filename extension. These programs can contain MS OS/2 *MULTIUSER* batch commands, CMD commands, MS OS/2 *MULTIUSER* utilities, and programs that can be run from the MS OS/2 *MULTIUSER* command interpreter, CMD.

Batch commands give you control over how your commands are carried out. They also let you send or display messages, set variables, and use replaceable parameters.

The MS OS/2 *MULTIUSER* batch commands are as follows:

Command	Purpose
CALL	Calls one batch program from another batch program.
ECHO	Turns the echo feature on or off.
ENDLOCAL	Restores the drive, directory, and environment settings that were in effect before the SETLOCAL command was used.
EXTPROC	Defines an external batch processor for a batch program.

Command	Purpose
FOR	Performs a command for a set of items.
GOTO	Causes processing to continue at the line after a specified label.
IF	Performs a command based on the result of a condition.
PAUSE	Temporarily suspends processing of the batch program.
REM	Adds remarks to a batch program.
SETLOCAL	Defines drive, directory, and environment variables that are used during batch program processing.
SHIFT	Changes the position of the replaceable parameters in batch-program processing.

The ECHO and FOR commands can be typed at the command prompt as well as placed in batch programs; all other batch commands can be used only in batch programs.

CREATING SIMPLE BATCH PROGRAMS

You create a batch program by typing commands in a file by using System Editor or any other text editor. Batch programs that you run from CMD, the MS OS/2 *MULTIUSER* command interpreter, must have the *filename* extension *.CMD.*

The examples in this discussion have been simplified as much as possible to illustrate batch programming commands and techniques. When you write a working batch program, however, you will probably want to embellish it with commands that provide error checking and commands that provide helpful information for you.

The simplest batch programs usually run utilities and display messages on the screen. They perform the same function that you would perform if you typed a series of commands at the command prompt.

For example, the following is a simple batch program called MYPROMPT, stored in the batch file *MYPROMPT.CMD*:

```
@ECHO OFF
REM This is a simple batch program
PROMPT The current time is $t$h$h$h$h$h$h$_$p
```

This program changes your prompt so that it displays the time, the current drive, and the directory. (The $h characters remove the seconds and hundredths of seconds from the time display.) To run this program, type the following:

```
MYPROMPT
```

The first line, "@echo off," is found in most batch programs. It prevents each batch command from being echoed on your screen as it is being processed by the command interpreter.

The second line, "REM This is a simple batch program," is a comment statement used to add notes to batch files. The command interpreter does not display REM statements when echoing is turned off, so you are free to add whatever comments you want to your batch programs.

The next line, "PROMPT," is the MS OS/2 *MULTIUSER* command you use to change the command prompt in the current session. By placing this command in a batch file, you only have to remember or look up the options once.

It is a good idea to remove blank lines from batch files, because each blank line will cause the command prompt to be displayed when the program is run.

Setting Up Your Environment with a Batch Program

One common way to use batch programs is to set up your environment after you start up CMD. For example, you can set environment variables such as PATH and PROMPT in an initialization file.

Suppose you want to create a system initialization file called *OS2INIT.CMD.* You want to set the MS OS/2 *MULTIUSER* session environment variable TMP in this file, as well as switch to a directory named WORK and start the SP spreadsheet program in the current session. *OS2INIT.CMD* would look like this:

```
@ECHO OFF
SET TMP=C:\TMP
CD \WORK
START /F SP
```

You could type these commands at the command prompt each time you login, but putting them in a batch file will save you time and effort.

Displaying a Message with a Batch Program

The following example creates a batch program that displays the message, "Use echo to display a message." Use System Editor or another text editor to create a file named *SPECIAL.CMD*, and then type the following:

```
@ECHO OFF
REM This program displays a message
ECHO Use echo to display a message.
```

You can use the REM batch command to add comments to your program. Any text after REM that is on the same line is treated as a comment. If the ECHO feature is on, REM commands are displayed as your program runs; if echoing is turned off (as in this example), no text is displayed.

The @ECHO OFF command turns off the ECHO feature (the default setting is on). This prevents the commands from being displayed on the screen as they are processed. The at sign (**@**) prevents the string "echo off" from being displayed as it turns the feature off. The @ECHO OFF command is often used at the beginning of batch programs to prevent unnecessary output from cluttering up your screen.

NOTE: When you are running a batch program in an MS OS/2 *MULTIUSER* session, you can also turn echoing off by typing the /Q option as an argument to the command that starts your batch program. For this reason, do not use /Q as an argument to any command in your batch file.

When echoing is off, you can use the ECHO command to display the text that follows it on the same line. You might want to display messages in your batch program to inform yourself or other users of what is taking place as the program runs or to inform them of error conditions that have occurred.

Creating a Custom Command

After using MS OS/2 *MULTIUSER* for a while, you might find there are commands and utilities you use on a daily basis. You can create custom versions of these commands to incorporate the options that you always use. As an example, suppose you work in a complex multilevel directory structure that contains many subdirectories and that you must return to your home directory several times a day.

To help you do this, you can create a customized version of the CD command using the HOME environment variable. The %HOME% environment variable specifies your home directory and is set up by the system when you login. Your version will display the message, "Returning to the home directory," switch you to your home directory, and display the directory list for the home directory one screenful at a time. To set up this special version of the CD command, type the following lines in a file called *HOME.CMD* in a directory in the path.

```
@ECHO OFF
ECHO Returning to the home directory
C:
CD %HOME%
DIR /P
```

Now, no matter what drive or directory you are in, you only need to type HOME and MS OS/2 *MULTIUSER* will return you to your home directory.

Performing a Series of Tasks with a Batch Program

A common use of batch programs is to perform copying operations to update files or to create backup disks. Suppose that you have specific files in particular directories on a floppy disk that are updated on a weekly basis, and that they then need to be copied to particular directories on your fixed disk. Suppose further that the specific files you are interested in are all of the files that contain the extensions *.EXE* and *.LIB* and all of the files in your INCLUDE directory, plus the file *README.DOC* from the root directory. This copying operation involves the following steps:

1. Copy all the *.EXE* files that are in the NEW directory on your floppy disk to the BIN directory located in your home directory.

2. Copy all the *.LIB* files in the NEW directory on your floppy disk to the LIB subdirectory of the BIN directory located in your home directory.

3. Copy all the files in the INCLUDE directory on your floppy disk to the INC directory located in your home directory.

4. Copy the *README.DOC* file from the root directory on your floppy disk to your home directory.

NOTE: You may need to RESERVE the floppy disk drive.

You can create a batch program that performs each of these steps by typing the following in a file named *TRANSFER.CMD*:

```
@ECHO OFF
REM This batch program copies files from drive A to drive C
ECHO Copying files ...
COPY A:\NEW\*.EXE C:\USR\DAVIDH\BIN
COPY A:\NEW\*.LIB C:\USR\DAVIDH\BIN\LIB
COPY A:\INCLUDE\*.* C:\USR\DAVIDH\INC
COPY A:\README.DOC C:\USR\DAVIDH\README.DOC
```

The COPY commands in the batch file perform the same operations as they would if they were typed on the command line. In this example, the exact files and directories to copy have been specified in the batch file. In later sections, you will see how you can type in arguments to a batch program.

To run this program, type TRANSFER and press **ENTER**. The *.CMD* extension is optional. You will see the COPY command display the files being copied as the copying operation progresses.

```
Copying files ...
A:\NEW\MOVE.EXE
A:\NEW\TEST.EXE
       2 file(s) copied.
A:\NEW\SLIBCE.LIB
A:\NEW\MLIBCE.LIB
A:\NEW\LLIBCE.LIB
       3 file(s) copied.
A:\INCLUDE\OS2.H
A:\INCLUDE\TYPES.H
A:\INCLUDE\STDIO.H
A:\INCLUDE\OS2DEF.H
       4 file(s) copied.
A:\README.DOC
       1 file(s) copied.
```

USING ADVANCED BATCH FILE TECHNIQUES

In addition to the basic tasks you can perform by using a batch program, there are more advanced techniques that will make your programs more flexible and powerful. For example, you can program your batch file so that it accepts parameters on the command line. You can also set conditions within batch files so that tasks are performed when those conditions are met. This section describes these advanced batch file techniques.

Using Replaceable Parameters

The TRANSFER program used in the preceding section illustrates how one program can perform several copying operations. These operations are fixed, since there is no way to specify which files are copied. But you can also write a batch program that allows you to specify which files to copy.

Batch programs can contain a special symbol called a replaceable parameter. This symbol is a percent sign (%) followed by a digit from 0 through 9. The replaceable parameters in a batch program are assigned values when you type the command and options. For example, suppose you type the TRANSFER command followed by three arguments, as follows:

```
TRANSFER \BIN \LIB \INC
```

If you use the %0, %1, %2, and %3 parameters in your batch file, %0 will be replaced by "TRANSFER", %1 by "\BIN", %2 by "\LIB", and %3 by "\INC".

The TRANSFER program created earlier can be changed so that you can specify on the command line whether you want the files from the BIN, INC, or LIB directories to be copied from a floppy disk in drive A to your fixed disk. To accomplish this, you will need three features: the IF batch command, the GOTO batch command, and the PAUSE batch command.

Using the IF Batch Command

The IF batch command causes a command to be carried out if a certain condition is true. The IF command can have one of the following forms:

IF *string1==string2 command*

Compares the character string *string1* with *string2*. If they are the same, *command* is processed.

IF EXIST *filename command*

Checks whether the named file exists. If the file does exist, *command* is processed.

IF ERRORLEVEL *number command*

Checks the exit code of the command or program that was run just before the IF statement. If the command or program returns an exit code that is equal to or greater than the *number* specified, *command* is processed. (For information about exit codes, see "Checking the Error Level" later in this chapter.)

The IF NOT batch command causes a command to be carried out if a certain condition is not true. For example, IF NOT EXIST *filename command* processes *command* if the *filename* does not exist.

Using the GOTO Batch Command

The GOTO batch command directs your program to switch to a different part of the file and continue processing the commands at that point. The GOTO command has the following form:

GOTO *label*

You tell the program what part of the file to switch to by using a *label*. The *label* is any string of characters (which may or may not form a word) that you put in the file, on a line by itself, at the point where you want your program to continue processing commands. For example, GOTO END will cause processing to continue at the command found after the label ":END." You identify a label in the batch file by typing a colon (:) in front of it.

Using the PAUSE Batch Command

The PAUSE batch command temporarily stops your program from running and displays the message, "Press any key when ready . . . ."

You can continue running the program by pressing a key, or you can quit the program by pressing **CTRL+C**.

Modifying the TRANSFER Batch Program

The following example shows the usage of the batch commands just described. As you see here, you can modify the TRANSFER program so that the first argument typed on the command line is used in the batch file:

```
@ECHO OFF
REM This batch program lets you decide
REM which directory to copy
ECHO Copying files from %1
PAUSE
IF "%1"=="" GOTO END
IF %1==\BIN GOTO COPYBIN
IF %1==\INC GOTO COPYINC
IF %1==\LIB GOTO COPYLIB
ECHO %1: Unknown option
GOTO END

:COPYBIN
COPY A:\BIN\*.* C:\USR\DAVIDH\BIN
GOTO END
:COPYINC
COPY A:\INC\*.* C:\USR\DAVIDH\INC
GOTO END
:COPYLIB
COPY A:\LIB\*.* C:\USR\DAVIDH\LIB
:END
```

The %1 parameter is used in the ECHO command to display whatever argument you type. The PAUSE command lets you determine whether to continue copying or quit the program by pressing **CTRL+C**.

The program uses the first IF command, IF "%1"=="" GOTO END, to check whether you have typed an argument. If no argument is found, the command causes the program to jump to the label ":END." If you do not specify an argument in an IF command, you must use quotation marks; if you are specifying an argument, the quotation marks are optional.

The next three IF commands cause the first argument that you type to be substituted for %1 and compared to the strings "\BIN," "\INC," and "\LIB." If a match occurs, the program jumps to the ":COPYBIN," ":COPYINC," or ":COPYLIB" label, respectively. After the program has copied the files, the GOTO command causes the program to jump to the ":END" label. Notice that if you type in an argument other than "\BIN," "\INC," or "\LIB" (that is, if none of the IF conditions were true), the message "Unknown option" is displayed and the program jumps to the ":END" label.

Specifying Multiple Arguments

To make your program more powerful, you will probably want to be able to specify more than one argument after the batch program. For example, the TRANSFER program that you created previously processes only the first argument typed. It ignores any additional arguments.

To specify two directories, you could modify the TRANSFER program to include the following lines:

```
IF %1==\BIN GOTO COPYBIN
IF %1==\LIB GOTO COPYLIB
IF %2==\BIN GOTO COPYBIN
IF %2==\LIB GOTO COPYLIB
```

Both %1 and %2 would be compared to the strings "\BIN" and "\LIB" and sent to the appropriate COPY command. But suppose that you want the option of typing up to 10 arguments after the TRANSFER command, with the arguments specified in any order. You would need many IF commands to test for every possible combination of arguments.

One way to do this is to use the SHIFT batch command. This command lets your program process any number of arguments that are typed in any order. Each time SHIFT is carried out, the program processes the next argument in the file. To see how this works, create a file named *DISPLAY.CMD* with the following lines:

```
@ECHO OFF
REM This displays a list of arguments
:DISPLAYARG
IF "%1"=="" GOTO END
ECHO %%1 IS NOW %1
SHIFT
GOTO DISPLAYARG
:END
```

Then type DISPLAY followed by a list of numbers for each argument. As it runs, the program will display each number on a line of its own. The statement if "%1"=="" checks to see whether all of the arguments have been read. In this example, two percent signs (**%%**) are used to display the actual percent character ("**%1**") instead of the %1 argument.

Now you can modify the TRANSFER program to accept any number of arguments, as follows:

```
@ECHO OFF
REM Copying multiple directories
PAUSE
:CHKARGS
ECHO Copying files from %1
IF "%1"=="" GOTO END
IF %1==\BIN GOTO COPYBIN
IF %1==\INC GOTO COPYINC
IF %1==\LIB GOTO COPYLIB
ECHO %1: Unknown option
GOTO END
:COPYBIN
COPY A:*.EXE C:\USR\DAVIDH\BIN
SHIFT
GOTO CHKARGS
:COPYINC
COPY A:*.INC C:\USR\DAVIDH\INC
SHIFT
GOTO CHKARGS
:COPYLIB
COPY A:LIB C:\USR\DAVIDH\LIB
SHIFT
GOTO CHKARGS
:END
```

The program compares each argument that you type to the strings "\BIN," "\LIB," and "\INC," as it did before. But the goto statements now direct the program to return to the ":CHKARGS" label after each argument is processed. Thus, you can type any number of valid arguments in any order and the effect is the same.

Repeating a Task

Sometimes you may want to perform a task on a set of different elements. For example, you may want to type out the contents of three files, or you may want to copy several directories from a floppy disk to a directory on your fixed disk. The FOR statement lets you specify the set of elements that a command will work with. To see its use, type the following lines in a file called *COUNT.CMD*:

```
@ECHO OFF
FOR %%X IN (One Two Three) DO ECHO %%X
```

If you now type COUNT, the following appears:

```
One
Two
Three
```

The FOR command causes the batch processor to substitute each element in the parentheses, one at a time, for "%%X." (You can use any single character in place of X.) The program processes whatever command appears after the DO command.

In the preceding example, each element in the set (One Two Three) is substituted for "%%X," then "%%X" is echoed to the screen. So the first time through the program, "One" is displayed; the second time, "Two" is displayed; and the third time, "Three" is displayed. Thus, the command takes the place of typing the ECHO command three times. In a batch file, you must use two percent signs (**%%**) in front of the X to distinguish a FOR variable from a replaceable parameter.

The following batch file, called *BINCOPY.CMD*, allows you to copy the contents of up to three directories that are in the current directory to your BIN directory on drive C:

```
@ECHO OFF
REM Updating %1 %2 %3
FOR %%X IN (%1 %2 %3) DO IF NOT EXIST %%X GOTO END
FOR %%X IN (%1 %2 %3) DO COPY %%X C:\BIN
:END
```

The first FOR command checks to see whether all of the directories exist. It does this by first substituting the arguments that you type for %1, %2, and %3, and then it uses the IF NOT EXIST command after the DO command to check to see that all of the directories do exist. If a directory does not exist, the program ends. The second FOR command does the actual copying by substituting each argument, one at a time, for the "%%X" in the COPY command.

Calling Another Batch Program

You can run, or call, one batch program from another batch program by using the CALL command. When the batch processor encounters this command, it passes control to the "called" program. When the program is finished running, control returns to the original program.

NOTE: If you start a second batch program from a batch program without using CALL, the second program will run. However, control will not return automatically to the original batch program after the second program finishes running. You must use the CALL command if you want control to be passed back to the original program.

For example, a main batch program could call other batch programs to perform the actual operations. The following short batch program calls the batch file *COPY1.CMD* if /C is typed as an argument, or it calls the batch file *DEL1.CMD* if /D is typed as an argument:

```
@ECHO OFF
REM This batch file calls other batch files
IF %1==/C CALL COPY1
IF %1==/D CALL DEL1
```

The batch files *COPY1.CMD* and *DEL1.CMD* perform the actual copying or deleting and then return control to the original batch file.

You cannot use pipes or redirection symbols with the CALL command. A batch file can call itself, but it should contain a command that ends it so that you do not remain in an infinite loop.

You have the option of using variables in your batch programs. These variables are saved in your environment just like the PATH environment variable. To set a variable in your batch program, use the SET command. For example, to set the variable STDERR to NUL, include the following line in your batch program:

```
SET STDERR=2^>NUL
```

Notice that in a batch file you must use the escape character (^) before the redirection symbol (>). The escape character (which is described later in this chapter) tells the batch processor to treat the greater-than symbol as a character, rather than processing it as a redirection symbol. Thus, the string "2>NUL" is simply assigned to STDERR without being processed.

If you type SET after running the program, you will see that STDERR has been saved in your environment. To prevent it from being saved, you can type the following to delete the setting:

```
SET STDERR=
```

There are several ways to use variables in batch files. Suppose you need to create a master phone list every week from various phone list files in different directories. You want to be able to just type the names of the files that contain the phone lists and have them collected in a file called *PHONE.LST* for later viewing. You decide that you need a batch file that will delete the old phone list, then create a new phone list that contains any phone files you specify on the command line. The batch file will then discard any error messages and display the new phone list on the screen.

```
@ECHO OFF
REM This batch file creates a new phone list
DEL PHONE.LST
:NEWPHONE
IF "%1"=="" GOTO DISPLAYIT
SET STDOUT=^>^>PHONE.LST & SET STDERR=2^>NUL
TYPE %1 %STDOUT%
SHIFT
GOTO NEWPHONE
REM Now display the phone list
:DISPLAYIT
TYPE PHONE.LST %STDERR%
```

When you run this batch file, STDOUT is replaced by the string ">>PHONE.LST," which appends output to the file PHONE.LST. So when the type %1 %STDOUT% command is run, %1 is replaced by the first argument typed, and the contents of the file specified are appended to the *PHONE.LST* file.

When the phone list is displayed, STDERR is replaced by the string "2>NUL," which redirects error messages encountered while sending the output of the file to NUL. Notice that the ampersand (&) (which is described later in this chapter) separates the two operations.

As before, STDOUT and STDERR are saved as variables in your environment after the program ends. You could assign STDOUT or STDERR to null values after you run the batch program, or you could set variables that are local to your batch program by using the SETLOCAL and ENDLOCAL batch commands. Just place SETLOCAL at the beginning of the program and ENDLOCAL at the end of the program. This tells the batch processor that any variables you set are active only while the batch program is running. After you run your program, these variables are not saved in your environment.

To see how these commands work, look at the batch file *LOCAL.CMD*:

```
@ECHO OFF
SETLOCAL
REM This program keeps the variables STDERR and MYVAR
REM from being saved in your environment.
IF "%1"=="" ECHO Valid arguments /S /M & GOTO END
IF %1==/S SET STDERR=ERRFILE
ID %1==/M SET MYVAR="Hi there!"
SET
:END
ENDLOCAL
```

This program assigns the variable STDERR to the file ERRFILE (if you type /S), and assigns the variable MYVAR to the string "Hi there!" (if you type /M). It then carries out the SET command, which displays your environment settings. You will see that either STDERR or MYVAR appears in the list of environment variables. However, if you type SET after running the batch program Local, STDERR or MYVAR will not appear as environment variables (unless you have set one of them previously).

Checking the Error Level

Some MS OS/2 *MULTIUSER* commands and utilities return an exit code when they are completed; that is, they send a number back to the system to tell it whether the program ran successfully or not. If the command or utility ran successfully, it returns an exit code of zero; if it did not run successfully, it returns an exit code greater than zero. You can check this value and specify what command will then run by placing the command IF ERRORLEVEL (or IF NOT ERRORLEVEL) immediately following the line that runs the original command or utility. The IF ERRORLEVEL command checks to see whether the exit code is equal to or greater than the number you specify.

For example, you could run the FORMAT utility in a batch file, check to see that the formatting operation was successful (returned an exit code of zero), and display a message on the screen by running the following program:

```
@ECHO OFF
FORMAT A:
IF ERRORLEVEL 1 ECHO An error occurred. & GOTO END
ECHO Format successful
:END
```

If the utility returns an exit code of greater than zero, an error message is displayed and processing jumps to the label ":END." Otherwise, the message "Format successful" is displayed.

Using Special Characters in a Batch Program

In "Running CMD," you learned how redirection and pipe symbols (<, >, >>, and |) could be used to specify more than one command on a line. The following symbols also have special meaning to MS OS/2 *MULTIUSER*:

Symbol	Meaning
^	Removes a special character's meaning.
( )	Groups commands.
&	Separates multiple commands.
&&	Performs one operation, then another.
||	Performs one operation or another.

While these special characters can all be used on the command line, they are most useful in batch programs. Each of these special characters is described in the following sections.

Removing a Special Character's Function

To remove the meaning of special characters, use the escape character (^) before the character. This lets you use special characters as regular characters in a string.

For example, to redirect the line "hello" into a file named GREET, type the following:

```
ECHO hello > GREET
```

If you use the escape character before the redirection symbol, the redirection symbol loses its meaning. To have the batch processor treat the redirection symbol as an ordinary character, type the following:

```
ECHO hello ^> GREET
```

Now the entire line is echoed to your screen:

```
hello > GREET
```

This is useful if you want to display a special character but do not want the symbol to perform any function.

Grouping Commands

To group commands, use parentheses [( )]. For example, to display the sorted output of a directory listing of drive A and the contents of the file *DIR.LST* on drive C, type the following:

```
(DIR A: & TYPE C:DIR.LST) | SORT
```

Combining Commands

To combine several commands, separate each command from the others with the ampersand (&). For example, to display a directory list for drive A, the version of MS OS/2 *MULTIUSER*, and the volume label of your current drive, type the following:

```
DIR A: & VER & VOL
```

MS OS/2 *MULTIUSER* processes the individual commands in order, from left to right.

Using the AND Operator

The AND operator (&&) processes the command to the left of the symbol. If the command is successful, it processes the command to the right of the symbol. If the first command is unsuccessful (that is, produces an error), the command to the right is not processed.

For example, the following command displays the contents of the file *BUDGET.FIL* only if the file exists in the current directory:

```
DIR BUDGET.FIL && TYPE BUDGET.FIL
```

Using the OR Operator

The OR operator (| |) processes either the command to the left of the symbol or the command to the right. If the command to the left is successful, it does not process the command to the right. If the command to the left is unsuccessful (that is, produces an error), the command to the right is processed.

For example, the following command either deletes the file called *LETTER.TMP* or displays a directory list of all files with the filename extension *.TMP*:

```
DEL LETTER.TMP | | DIR *.TMP
```

If *LETTER.TMP* exists, it is deleted. If *LETTER.TMP* does not exist, a directory list of all the files with the *.TMP* extension is displayed.

USING STARTUP FILES

Startup files are optional; that is, they are not required for system operations. Startup files are used primarily to tailor a your working environment. There are several types of startup files, each with a different scope. These files are listed below.

- *CONFIG.USR*, the startup file for a loginname.

  The *CONFIG.USR* file is a configuration file and may contain certain configuration commands. The commands within the file are processed when you login to the system and a new login is created. *CONFIG.USR* is useful for automatically starting programs and setting additional environment variables when a you login to the system. The *CONFIG.USR* file resides in your home directory. There may be a separate *CONFIG.USR* file for each user in the system.

  For example, if C:\USR\DAVIDH is the home directory for the *loginname* DAVIDH.WRKGRP, then C:\USR\DAVIDH\CONFIG.USR specifies the file.

- *STARTUP.CMD*, the startup file for the Program Selector.

  STARTUP.CMD is a batch file and may contain any valid batch file command. The *STARTUP.CMD* file is processed by the Program Selector. When the Program Selector starts, the Program Selector invokes CMD to process the *STARTUP.CMD* batch file. *STARTUP.CMD* is useful for automatically starting programs under the Program Selector and performing other batch operations when the Program Selector starts. The *STARTUP.CMD* file resides in the directory that is the current directory when you login to the system (working directory). There may be a separate *STARTUP.CMD* file for each user in the system.

  For example, if C:\USR\BRAD is the home directory for the loginname BRAD.MANAGER, then C:\USR\BRAD\STARTUP.CMD specifies the file.

- CMD /K [*batch filename*], the startup file option for CMD

  The /K option for CMD has a parameter which specifies the path and filename of a batch file that is executed when CMD starts. It is useful for setting up parameters for the session in which the *CMD.EXE* program executes.

The *CONFIG.USR* File

The *CONFIG.USR* file in your home directory uses configuration commands. The following commands are valid for the *CONFIG.USR* file.

REM
RUN
SET

Refer to the *Citrix MULTIUSER Command Reference* for more information on these commands.

Batch Files

The startup batch files, *STARTUP.CMD* and /K [*batch file*], may use any batch command or command. Batch files may not use configuration commands. Batch commands include the following:

CALL
ECHO
ENDLOCAL
EXTPROC
FOR
GOTO
IF
PAUSE
REM
SET
SETLOCAL
SHIFT

Refer to the *Citrix MULTIUSER Command Reference* for more information on these commands.

CHAPTER 7

USING Citrix *MULTIUSER* SYSTEM EDITOR

INTRODUCTION

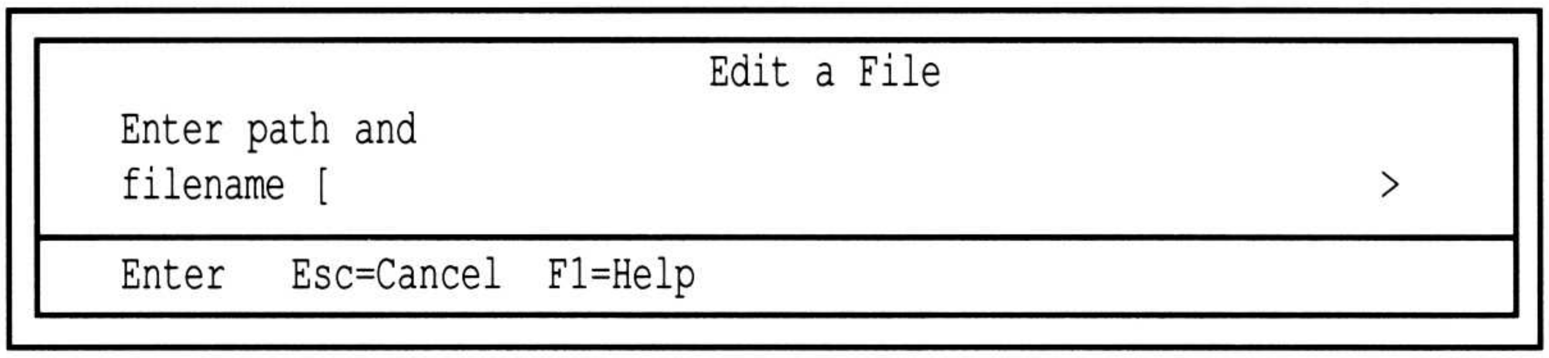
System Editor is the MS OS/2 *MULTIUSER* full screen text editor. It allows you to display and edit data files.

The MS OS/2 *MULTIUSER* System Editor works with the MS OS/2 *MULTIUSER* HPFS to preserve extended attributes and security attributes on existing files. It also allows you to edit long *filenames* permitted by HPFS.

STARTING AND QUITTING SYSTEM EDITOR

Starting System Editor

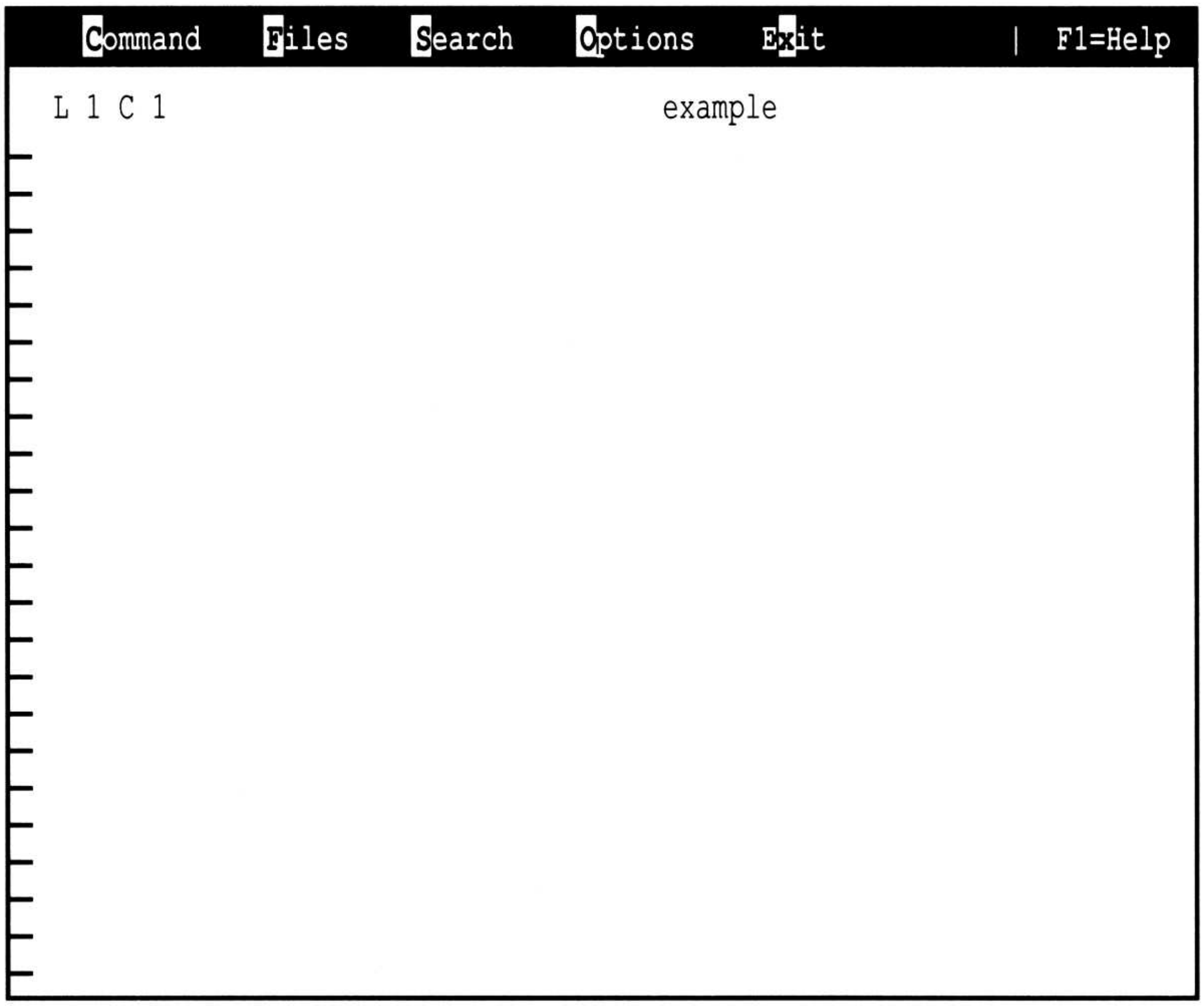
To start System Editor, type **E** at the command prompt. Your screen will clear and the System Editor screen will appear, showing the "Edit a File" text window.



```
                         Edit a File
  Enter path and
  filename [                                              >

  Enter   Esc=Cancel  F1=Help
```

Type the name of the file you want to edit, and press **ENTER**. If this file does not exist, System Editor creates it. If you are creating a new file, the System Editor screen looks like this:



You can also enter *filename(s)* on the command line. For example, to edit your *STARTUP.CMD* file using System Editor, type the following at the CMD prompt:

```
E STARTUP.CMD
```

This example starts System Editor and opens your *STARTUP.CMD* file.

You can also specify multiple *filenames* when you start System Editor from CMD. For example, type the following:

```
E STARTUP.CMD DATAFILE.1 DATAFILE.2
```

This example starts System Editor and opens each of the specified files. You can now switch between these files by using the **F8** and **ALT+F8** keys. For more information about switching between files, see "Working with Multiple Files" later in this chapter.

Quitting System Editor

To save your file and quit System Editor, use the Save and Exit This File command:

- Press **F10** to select the menu bar.
- Use the **LEFT** and **RIGHT** keys to select the Exit menu.
- Use the **UP** and **DOWN** keys to choose the Save and Exit This File command.
- Press **ENTER**.

To quit System Editor without saving your changes, use the Exit This File command:

- Press **F10** to select the menu bar.
- Use the **LEFT** and **RIGHT** keys to select the Exit menu.
- Use the **UP** and **DOWN** keys to choose the Exit This File command.
- Press **ENTER**.

If you have made changes to the file, a text window appears, telling you that the file has been modified and asking you to confirm that you want to quit System Editor without saving your changes. Type **Y** and press **ENTER** to quit without saving your changes; type **N** and press **ENTER** to continue using System Editor.

USING HELP

The System Editor has online Help information. You get Help by pressing **F1**. You can use Help any time while you're using System Editor.

The information that Help presents depends on what you have selected in System Editor. For example, if you have selected a menu, pressing **F1** gives you Help information for that menu; if you have selected a command on a menu, pressing **F1** gives you information for that command. If nothing is selected, you get a text window with general information about using Help.

Using the Help Index

Help also provides an online index to System Editor commands so that you can easily find out how a particular command works. To use the index, follow these steps:

1. From anywhere in System Editor, press **F1**. A Help text window appears.

2. Press **F11** (**ALT+F1** if your keyboard has only ten function keys). The Help index appears.

3. Use the **DIRECTION** keys to choose the command you want information about and press **ENTER**.

4. Press **ESC** to return to the index, or press **ESC** twice to quit Help.

You can also use Help to view the current System Editor key assignments. Use Help as follows:

1. From anywhere in System Editor, press **F1**. The Help text window appears.

2. Press **F9**. System Editor opens the *E.DEF* file and it appears on your screen.

3. You can scroll through the file to look at the various key definitions.

4. Press **F3** to close the file.

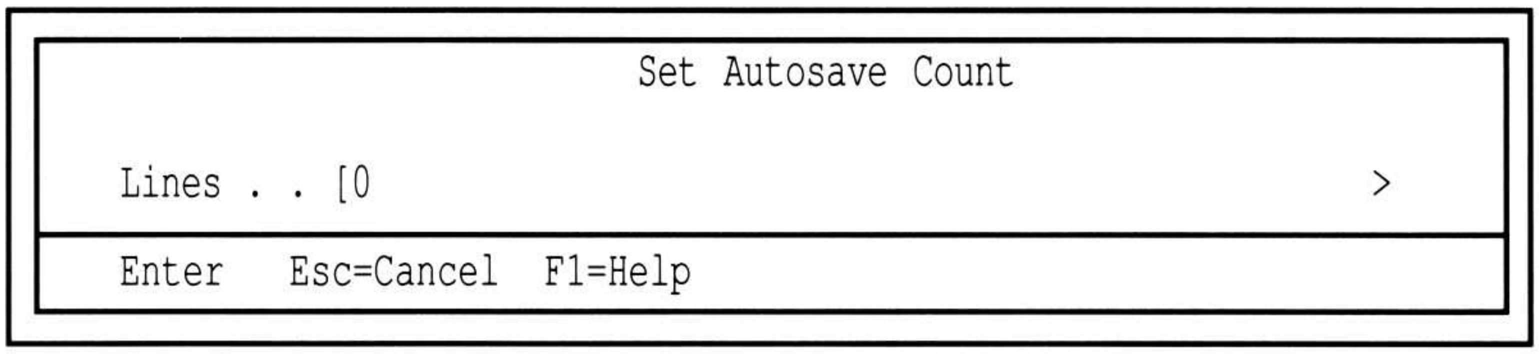
SAVING A FILE

You can save any changes you've made to a file by using the Save And Continue command from the Files menu or by pressing **F2**. This command saves any changes you've made to the current file.

By using the Autosave command from the Options menu, you can also tell System Editor to save your file automatically after a specified number of lines have been changed. The Autosave command creates a backup copy of the file you are editing, in case your system loses its power. If your system loses its power, the backup file is saved in the current directory and given a name of the following form: up to the first five letters of the filename, following by a three-digit, sequential number, followed by a numerical three-digit extension. For example, a backup copy of the file *MOLLY.TXT* might be named *MOLLY003.000* by the Autosave command.

To use the Autosave command, follow these steps:

1. Select the Options menu and choose the Autosave command. The following text window appears:



```
                       Set Autosave Count

 Lines . . [0                                                  >

 Enter   Esc=Cancel  F1=Help
```

2. In the Lines text box, type the number of lines to be changed before System Editor automatically saves your file.

3. Press **ENTER**.

You must still save the file when you quit System Editor if you want your changes saved in the original file rather than in a backup file.

TYPING AND FORMATTING TEXT

You can start typing as soon as you've opened a file in System Editor. The cursor moves to the right as you type. If your typing goes beyond the borders of the screen, System Editor automatically scrolls the text to the left so that the cursor is always visible.

Moving the Cursor

When you start System Editor, the cursor initially appears in the upper left corner. If you switch to another file and then come back to the first file, the cursor appears where you left it. You can move the cursor to wherever you want to insert or edit text.

To move the cursor, press the following keys:

To Move the Cursor	Press
Up one line	**UP**
Down one line	**DOWN**
To the left one character	**LEFT**
To the right one character	**RIGHT**
To the beginning of the current line	**HOME**
To the end of the current line	**END**
To the beginning of the file	**CTRL+HOME**
To the end of the file	**CTRL+END**

Inserting Text

If you're editing an existing file, you can insert new text into the file. Use the **INS** key to switch between insert and replace modes. The cursor appears as a flashing rectangle when you're in insert mode. As you insert new text, existing text moves to the right of the cursor.

When you start System Editor, you are in insert mode.

Replacing Text

You can replace existing text by typing over it. Use the **INS** key to switch between insert and replace modes. The cursor appears as a flashing line when you're in replace mode.

Formatting Text

You can type text exactly as you want it to appear, using the following keys to place the text where you want it:

To Do This	Press
Insert a space	**SPACEBAR**
Delete the character to the left of the cursor	**BACKSPACE**
Delete the character marked by the cursor	**DELETE**
End a line of text	**ENTER**
Indent a line one tab (in insert mode)	**TAB**
Insert a tab stop (in insert mode)	**TAB**
Move the cursor back one tab stop	**SHIFT+TAB**

To split a line of text, move the cursor to the beginning of the text that you want to move and press **ALT+S**. System Editor inserts a blank line and places the text on it.

To join two lines of text, move the cursor to the first of the two lines you want joined and press **ALT+J**. System Editor joins the line below with the current line.

SCROLLING

If the text in the file is longer or wider than can be shown at one time, you can scroll through the file to view the text.

You scroll the text of a file by using a **DIRECTION** key to move the cursor to the edge of the screen and then pressing the same **DIRECTION** key again. For example, to see the next two lines below the bottom of the screen, do the following:

1. Press the **DOWN** key until the cursor is at the bottom of the screen.

2. Press the **DOWN** key twice more to display the next two lines of text.

NOTE: When you scroll up or down, the text scrolls one line at a time. When you scroll left or right, however, the text scrolls half a screen at a time, even though the cursor moves only one column within the file.

You can scroll up or down by the screenful instead of by the line by using the **PAGE UP** and **PAGE DOWN** keys. You can scroll to the beginning or end of a long line by using the **HOME** and **END** keys.

EDITING A FILE

You edit a file with System Editor by using commands from menus, from the command line, or from the keyboard. You can delete text, move or copy text to a new location, and search for text within a file. If you change your mind after editing, you can cancel your last edit. You can even change the name of the file you're working on without quitting System Editor.

Deleting Text

Once you've typed text, you can delete it by using the following methods:

To Delete	Press
The character to the left of the cursor	**BACKSPACE**
The character marked by the cursor	**DELETE**
The current line	**CTRL+BACKSPACE**
From the cursor to the beginning of the line	**CTRL+B**
From the cursor to the end of the line	**CTRL+E**

You can also delete several lines of text at one time. To do this follow these steps:

1. Use the **DIRECTION** keys to move the cursor to the first line of text you want to delete.

2. Press **ALT+L** to mark the line of text. The line of text is now marked for deleting and is highlighted on the screen. If you change your mind, press **ALT+U** to remove the mark from the text.

3. Use the **DIRECTION** keys to move the cursor to the last line of text you want to delete.

4. Press **ALT+L** to mark the line of text. The line of text and every line between it and the first marked line are now marked for deletion and are highlighted on the screen. If you change your mind, press **ALT+U** to remove the mark from the text.

5. Press **ALT+D** to delete the lines of text.

Moving Text

In System Editor, you can move a line or lines of text from one place to another in a text file. Before you can move text, you must mark it. To move text, follow these steps:

1. Use the **DIRECTION** keys to move the cursor to the first line of text you want to move.

2. Press **ALT+L** to mark the line of text. The line of text is now marked for moving and is highlighted on the screen. If you change your mind, press **ALT+U** to remove the mark from the text.

3. If you want to move more than one line of text, use the **DIRECTION** keys to move the cursor to the last line you want to move and press **ALT+L** to mark the line of text. The line of text and every line between it and the first marked line are now marked for moving and are highlighted on the screen.

4. Move the cursor to where you want the text to appear (text will be moved to the line below the cursor).

5. Press **ALT+M** to move the line(s) of text.

Copying Text

If you want to use the same text more than once in a file, you can copy existing text to another place in the file. Before you can copy text, you must mark it. To copy text, follow these steps:

1. Use the **DIRECTION** keys to move the cursor to the line of text you want to copy.

2. Press **ALT+L** to mark the line of text. The line of text is now marked for copying and is highlighted on the screen. If you change your mind, press **ALT+U** to remove the mark from the text.

3. If you want to copy more than one line of text, use the **DIRECTION** keys to move the cursor to the last line you want to copy and press **ALT+L** to mark the line of text. The line of text and every line between it and the first marked line are now marked for copying and are highlighted on the screen.

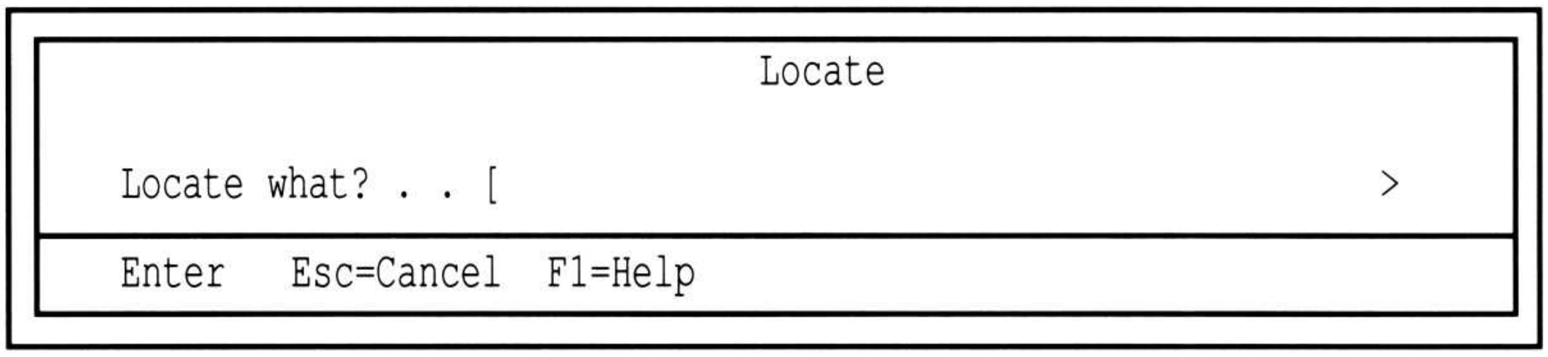
4. Move the cursor to where you want the copied text to appear (text will appear on the line below the cursor).

5. Press **ALT+C** to copy the line(s) of text.

Searching for Text

You can search for specific text in a file by using the Locate command from the Search menu. Follow these steps to search for text:

1. Select the Search menu and choose the Locate command. The following text window box appears:



```
                          Locate

  Locate what? . . [                                   >

  Enter   Esc=Cancel  F1=Help
```

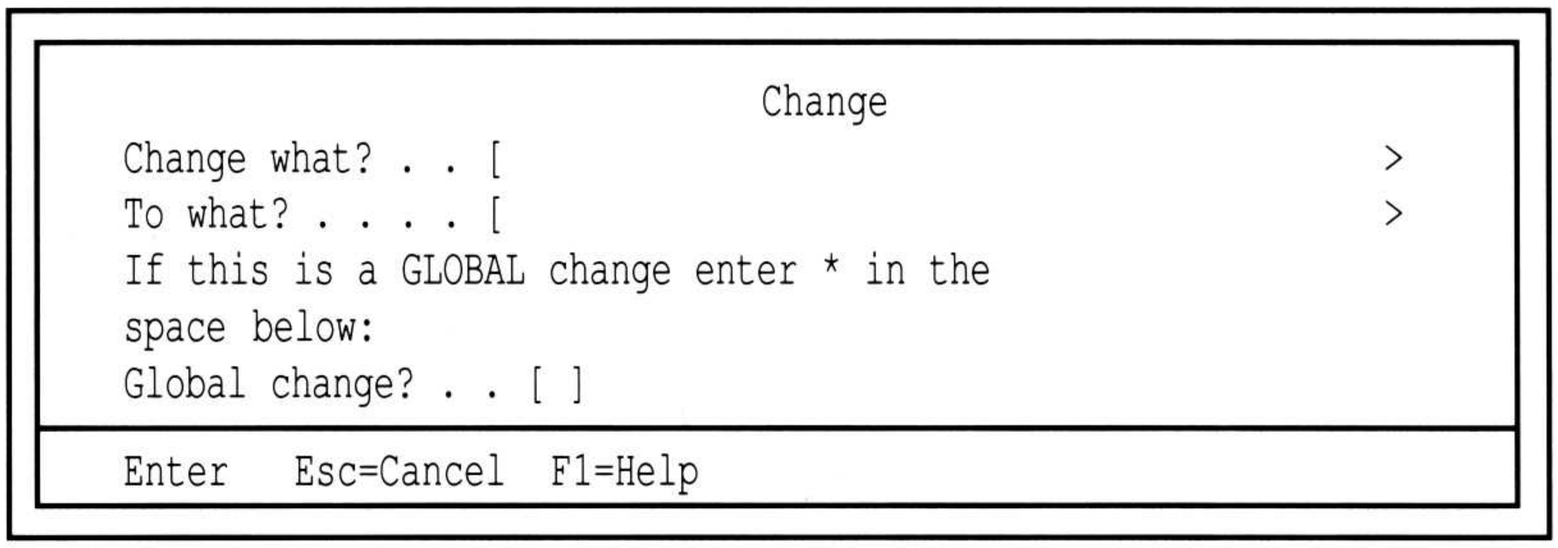
2. In the text box, type the text you want System Editor to find and press **ENTER**.

System Editor searches forward from the cursor to the end of the file. If you want System Editor to search the entire file, change the Searchwrap setting to ON by using the Searchwrap command from the Options menu.

To find further occurrences of the specified text, repeat the procedure.

You can also search for specific text and replace it with other text by using the Change command from the Search menu. To do this, follow these steps:

1. Select the Search menu and choose the Change command. The following text window appears:



```
                                Change
Change what? . . [                                        >
To what? . . . . [                                        >
If this is a GLOBAL change enter * in the
space below:
Global change? . . [ ]

Enter   Esc=Cancel  F1=Help
```

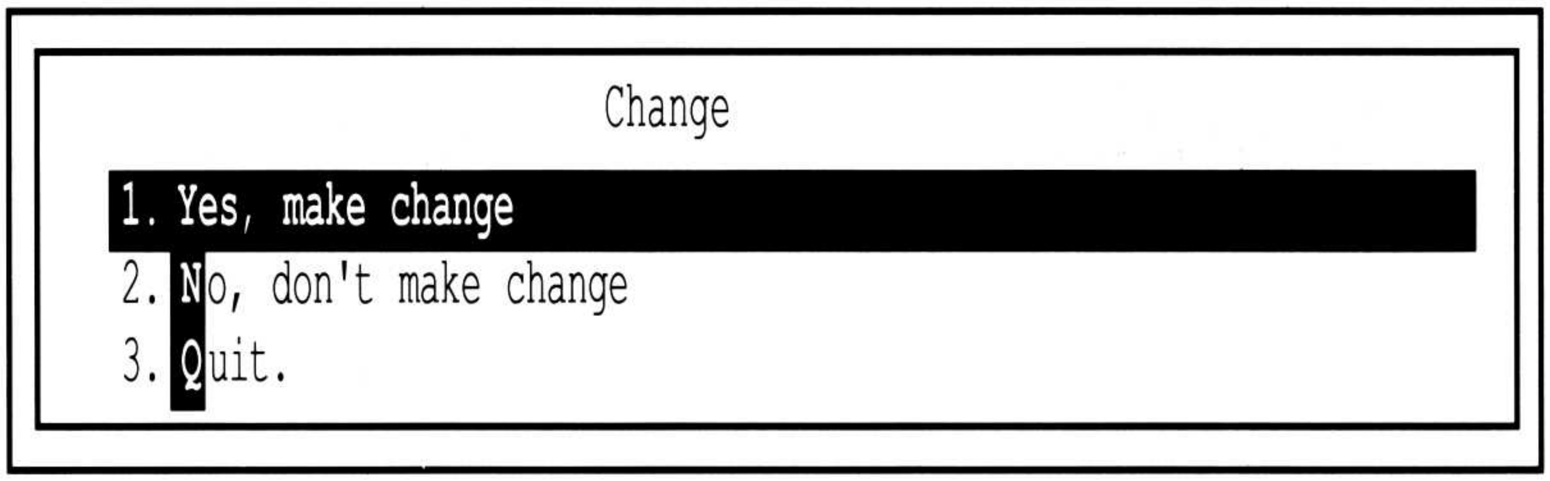
2. Type the text to be searched for in the Change What? text box. Use the **DOWN** key to move to the next text box.

3. Type the replacement text in the To What? text box.

4. If you want the replacement made for every occurrence of the text within the file, type an asterisk (∗) in the Global Change? text box.

5. Press **ENTER**.

6. If you specified a global change, the replacements are made. If you did not specify a global change and the specified text is found, the following text window appears:



```
                    Change
1. Yes, make change
2. No, don't make change
3. Quit.
```

7. Select the appropriate option and press **ENTER**.

Undoing an Edit

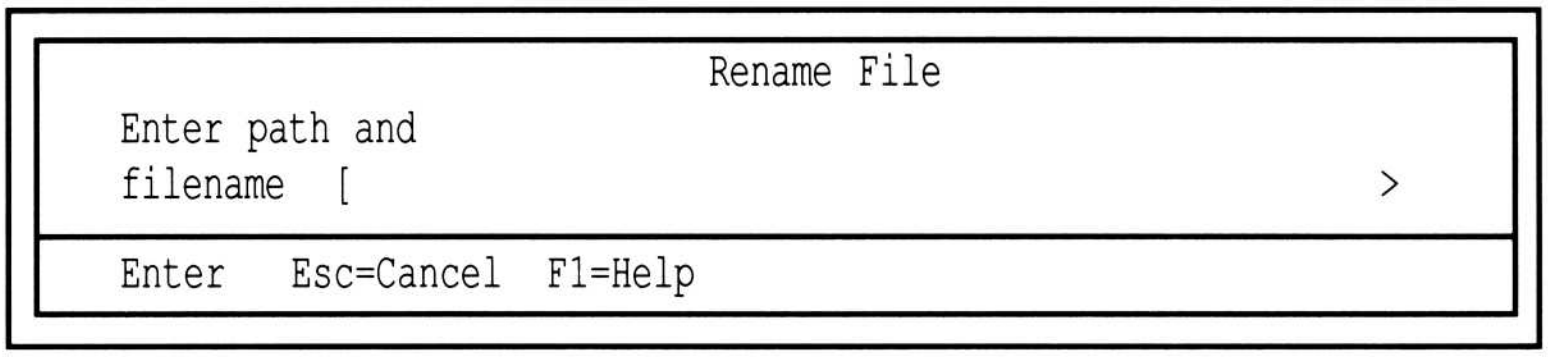
If you change your mind after editing a line and you haven't yet moved the cursor from that line, you can cancel your last edit by pressing **F9** to restore the line to the way it was before you changed it.

Pressing **F9** only works if your edit consisted of typing new text (in either insert or replace mode) or deleting one character.

Changing the Name of a File

You can change the name of the file you are working in by using the Rename A File command from the Files menu. To do this, follow these steps:

1. Select the Files menu and choose the Rename A File command. The following text window appears:



```
                         Rename File
Enter path and
filename  [                                              >

Enter   Esc=Cancel  F1=Help
```

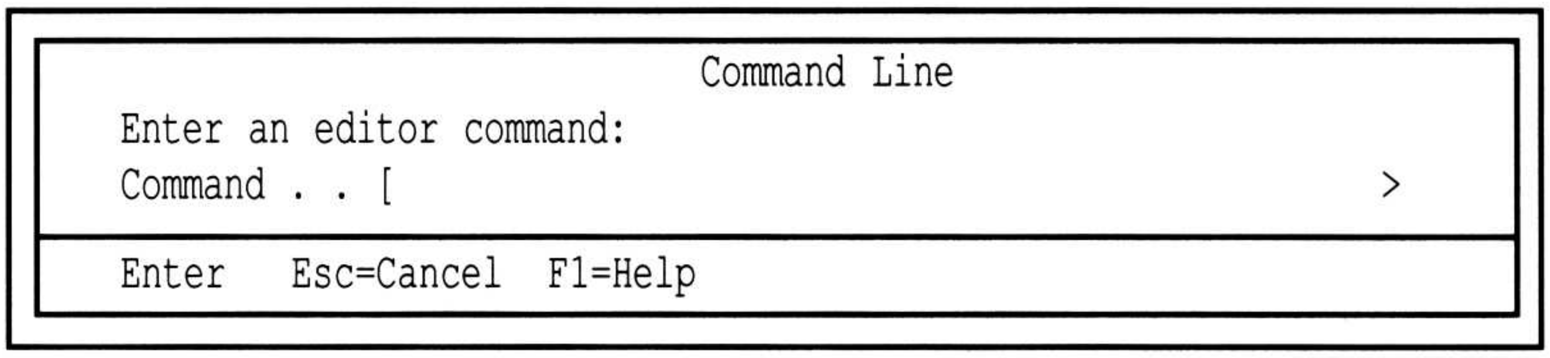
2. Type the new filename and press **ENTER**.

Notice that the new *filename* now appears in the title bar.

USING THE COMMAND LINE

While you can use function keys or key combinations to carry out some of the System Editor commands, you need to type other commands on the System Editor command line. To use the command line, do the following:

- Press **F5**. The Command Line text window appears.



```
                          Command Line
  Enter an editor command:
  Command . . [                                              >

  Enter   Esc=Cancel  F1=Help
```

From here you can type any of the System Editor commands.

Use Help for descriptions of the System Editor commands that you can type on the command line.

WORKING WITH MULTIPLE FILES

In System Editor, you can work with several files at a time. Suppose you are editing a file in System Editor. To open a second file for editing, without quitting the active file, follow these steps:

1. Select the Files menu and choose the Edit A File command.

2. Type the name of the file you want to work on and press **ENTER**.

You now have two files open. You can switch between these files by pressing **F8**, to switch to the next file, or pressing **ALT+F8**, to switch to the previous file. When only two files are open, these commands function exactly the same way: they allow you to switch back and forth between the two files.

If you start System Editor from CMD, you can specify more than one file on the command line. You can then switch between these files by using the **F8** and **ALT+F8** keys.

To start System Editor and switch between files, follow these steps:

1. At the CMD prompt, type

   E *filename1 filename2 filename3*

   and press **ENTER**. The file specified by *filename1* appears on your screen.

2. Press **F8**. The file specified by *filename2* now appears on your screen.

3. Press **F8**. The file specified by *filename3* now appears on your screen.

Pressing **F8** takes you through the files in the order in which you specified them on the command line. You can go backwards through the order by pressing **ALT+F8**.

Displaying the Current Directory

You can display the current directory without quitting System Editor. You do this by pressing **CTRL+D**. You may want to do this to verify filenames before specifying them in System Editor.

Merging Files

You can merge the contents of a file with the file you are working on. To do this, follow these steps:

1. Move the cursor to the line just above where you want the merged text to be placed.

2. Select the Files menu and choose the Merge A File command. A text window appears, prompting you for the *filename.*

3. Type the name of the file to be merged and press **ENTER**. If the file is in another directory, type the directory path and *filename.*

The merged text now appears just below the cursor.

DRAWING A BOX

In System Editor, you can draw boxes within your file by drawing lines and adding corners. These boxes are used for enhancing the way your file looks on the screen. Use the following key combinations to draw boxes:

To Draw	Press
The top-left corner	**CTRL+F1**
The top-right corner	**CTRL+F2**
The bottom-left corner	**CTRL+F3**
The bottom-right corner	**CTRL+F4**
A horizontal bar	**CTRL+F5**
A vertical bar	**CTRL+F6**

ASSIGNING YOUR OWN KEY DEFINITIONS

In System Editor, you can assign your own key definitions by modifying the key-definition file, *E.DEF*. The key-definition file is a data file that tells System Editor how to interpret a particular keystroke.

System Editor will use the *E.DEF* file in the current directory. If *E.DEF* is not in the current directory, System Editor uses PATH to locate your *E.DEF*. If none is found, a default set of key definitions is used.

For example, to assign your own key definitions, follow these steps:

1. Using System Editor, open the *E.DEF* file.
2. Using the **DIRECTION** keys, scroll through the file to locate the key definition that you want to change and make the change.
3. Save the *E.DEF* file and quit System Editor.

Each of the definition (def) statements defines a particular keystroke. The name of the key or key combination is to the left of the equal sign (=). The command to the right of the equal sign (=) determines the key function. The number sign (#) is a comment symbol and the text to the right of it describes the function.

For example, suppose you want to change the key assignment for deleting the current line of text from **CTRL+BACKSPACE** to **CTRL+L**. To do this, follow these steps:

1. Using System Editor, open the *E.DEF* file.
2. Select the Search menu and choose the Locate command.

3. Type DELETE CURRENT LINE in the Locate What? text box and press **ENTER**. System Editor takes you to the following line:

   DEF C_BACKSPACE = DELETE # DELETE CURRENT LINE

4. Change C_BACKSPACE to C_L.

5. Save the *E.DEF* file and quit System Editor.

Now when you use System Editor, you'll press **CTRL+L** to delete the current line of text.

USING MS OS/2 *MULTIUSER* COMMANDS

You can also use MS OS/2 *MULTIUSER* commands from the System Editor command line. With this feature, you can use an MS OS/2 *MULTIUSER* command without quitting System Editor. To use an MS OS/2 *MULTIUSER* command, follow these steps:

1. Press **F5** to use the System Editor command line.

2. In the Command text box, type DOS, followed by the command you want to use and press **ENTER**. For example, if you want to list the directory contents of the root directory on drive D, type DOS DIR D:.

When the command is complete, you return to the System Editor screen.

NOTE: Although you type DOS on the System Editor command line, the commands that you can run from System Editor are MS OS/2 *MULTIUSER* commands, not MS DOS commands. Since System Editor runs in a full-screen MS OS/2 *MULTIUSER* session, you cannot run MS DOS commands from it.

APPENDIX A

TERMINALS USED WITH Citrix *MULTIUSER*

INTRODUCTION

This appendix provides specific information on operating some of the terminals that can be connected to an MS OS/2 *MULTIUSER* system.

SOME GENERAL CONSIDERATIONS

Most terminals have brightness and contrast controls, usually located on the front or the side of the terminal. These controls are adjustable for comfortable viewing. The terminal power switch is also usually located on the front or side of the terminal. Most terminals will beep when turned on to indicate that they have started up properly. If a terminal ever fails to come up properly or displays garbled data on the screen after power-up, contact your System Administrator.

USING TERMINALS

The sections that follow give specific operating information for the various terminals.

ADDS 2025 Terminal

The ADDS 2025 terminal supports 25-line modes only. The power switch is on the right side of the video element near the back. There are sliding controls for contrast and brightness on the right side of the video element to the front of the power switch. Plug the cable to the MS OS/2 *MULTIUSER* system into the connector marked SES1-EIA.

NOTE: **CTRL+BREAK** does not work on the ADDS 2025 terminal. Use **CTRL+C**.

Esprit OPUS 3n1+ Terminal

The Esprit OPUS 3n1+ terminal supports 25-line modes only. The power switch is on the front of the video element on the right. A brightness control is hidden behind a door on the front of the video element to the left. The contrast control is a black shaft on the rear of the video element. Plug the cable to the MS OS/2 *MULTIUSER* system into the connector marked MAIN PORT.

IBM 3151 Terminal (Models 51/61)

The IBM 3151 terminal supports 25-line modes only. The power switch is on the front of the video element on the right. The brightness control is on the front of the video element to the left of the power switch. Plug the cable to the MS OS/2 *MULTIUSER* system into the rightmost connector on the back of the video element. Note that only IBM 3151 Models 51 and 61 are supported.

Kimtron KT-70PC Terminal

The Kimtron KT-70 terminal supports 25-line modes only. The power switch is on the base of the video element on the right. The brightness control is on the lower left front corner of the video element. Plug the cable to the MS OS/2 *MULTIUSER* system into the main port (J1) connector on the back of the video element.

LINK MC2PC Terminal

The LINK MC2PC terminal supports 25-line modes only. The power switch is on the right side of the video element near the front. The brightness and contrast controls are located on the underside of the front of the video element. Plug the cable to the MS OS/2 *MULTIUSER* system into the connector marked MAIN on the back of the video element.

LINK MC5 Terminal

The LINK MC5 terminal supports 25-line modes only. The power switch is on the right side of the video element near the front. The brightness and contrast controls are located on the underside of the front of the video element. Plug the cable to the MS OS/2 *MULTIUSER* system into the connector marked MAIN on the back of the video element.

Televideo 965 Terminal

The Televideo 965 terminal supports 25 and 43-line modes. The power switch is on the front of the video element on the right. The sliding brightness control is immediately to the left of the power switch on the front of the video element. Plug the cable to the MS OS/2 *MULTIUSER* system into the connector marked MAIN on the back of the video element.

NOTE: **CTRL+BREAK** does not work on the Televideo 965 terminal. Use **CTRL+C**.

Wyse 60 Terminal

The Wyse 60 terminal supports 25 and 43-line modes. The power switch is on the right of the video element near the front. The sliding brightness control is on the front of the video element on the right. Plug the cable to the MS OS/2 *MULTIUSER* system into the connector marked MODEM on the back of the video element.

Wyse 150 Terminal

The Wyse 150 terminal supports 25-line modes only. The power switch is on the right of the video element near the back. The rotating brightness controls are on the right of the video element to the front of the power switch. Plug the cable to the MS OS/2 *MULTIUSER* system into the connector marked MAIN on the back of the video element.

INDEX

Citrix Systems, Inc.
210 University Drive, Suite 700, Coral Springs, FL 33071: Phone (305) 755-0559